C. GALAIS

PASSEPORT POUR COMMODORE 64

POCHE *informatique* 10

Collection
POCHE-INFORMATIQUE

PASSEPORT POUR COMMODORE 64

par
Claudy GALAIS

Diffusion :
ÉDITIONS TECHNIQUES ET SCIENTIFIQUES FRANÇAISES
2 à 12, rue de Bellevue, 75940 PARIS CEDEX 19

COLLECTION E.T.S.F. MICRO-SYSTEMES

1 - A. VILLARD et M. MIAUX, *Un microprocesseur pas à pas*
2 - A. VILLARD et M. MIAUX, *Systèmes à microprocesseur*
3 - P. GUEULLE, *Maîtrisez votre ZX 81*
4 - E. FLOEGEL, *Du Basic au Pascal*
5 - P. COURBIER, *Vous avez dit Basic ?*
6 - M. MARCHAND, *Vous avez dit micro ?*
7 - P. GUEULLE, *Pilotez votre ZX 81*
8 - M. JACQUELIN, *La micro-informatique et son A-B-C-*
9 - M. OURY, *Maîtrisez les TO 7 et TO 7-70*
10 - P.GUEULLE, *Pilotez votre Oric (1 + ATMOS)*
11 - P. JOUVELOT et D. LE CONTE DES FLORIS, *Système d'exploitation et logiciel de base des micro-ordinateurs*
12 - P. GUEULLE, *Robotisez votre ZX 81*

COLLECTION POCHE-INFORMATIQUE

1 - G. ISABEL, *50 programmes pour ZX 81*
2 - P. GUEULLE, *Montages périphériques pour ZX 81*
3 - C. GALAIS, *Passeport pour Applesoft*
4 - R. BUSCH, *Passeport pour BASIC*
5 - M. ROUSSELET, *Mathématiques sur ZX 81*
6 - C. GALAIS, *Passeport pour ZX 81*
7 - G. PROBST, *50 programmes pour Casio FX-702 et 801 P*
8 - G. PROBST, *60 programmes pour Casio PB-100*
9 - M. SAAL, *utilitaires pour ZX 81*
10 - C. GALAIS, *Passeport pour Commodore 64*

« La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que « les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toutes représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'Art. 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les Art. 425 et suivants du Code pénal ».

© 1984 — E.T.S.F.

ISBN 2-85535-081-6 ISSN 0757-6730

SOMMAIRE

Récapitulatif des fonctions, commandes et instructions utilisées par le COMMODORE 64.

ABS	Donne la valeur absolue d'un argument	9
"+"	Opère la somme de deux éléments	10
"="	Affecte l'élément de droite à celui de gauche.	11
AND	Opérateur de relation logique ET	12
ASC	Donne le code ASCII	14
ATN	Donne l'arc tangente d'un argument radians.	15
CHR$	Donne le caractère ASCII	17
CLOSE	Ferme un fichier	18
CLR	Vide les variables	19
CLR/HOME	Efface l'écran	20
CMD	Choisit un périphérique en sortie	21
CONT	Continue un programme arrêté par STOP	22
COPY	Copie un fichier	23
COS	Donne le cosinus d'un argument radians	24
DATA	Entre des données	25
DEF FN	Affecte une fonction à une variable	27
" < > "	Permet d'effectuer une comparaison	28
DIM	Réserve la place nécessaire à un tableau	29
"/"	divise un élément par un autre	30
"$"	Définit une variable alphanumérique	31
DUPLICATE	Copie une disquette sur une autre	32
E	Transforme les grands nombres en exposants	33
EGAL	Effectue une comparaison	35
END	Arrête le déroulement d'un programme	36
ERREUR	Indique un type de faute de programmation	37
EXP	Elève la constante E (2.7182818) à une puissance	41
FN	Fonction définie par DEF FN	42
FOR	Affecte une variable de contrôle	43

FRE	Donne le nombre d'octets disponible	45
GET	Affecte à une variable le caractère tapé	46
GET #	Prend lettre par lettre des données sur fichier	47
GOSUB	Envoie à un sous-programme	48
GOTO	Fait un saut à une ligne spécifiée	49
"“	Délimitent une chaîne de caractères	50
IF	Teste une condition	51
" < "	Compare deux valeurs ou deux chaînes	53
" < ="	Compare deux valeurs ou deux chaînes	54
INITIALIZE	Remet le drive à l'état initial	55
INPUT	Affecte des variables	56
INPUT #	Affecte des variables à partir d'un fichier	57
INT	Donne l'entier d'un nombre	58
LEFT$	Extrait des caractères gauches	59
LEN	Donne la longueur d'une chaîne	60
LET	Affecte une chaîne, ou une valeur à une variable	61
LIST	Affiche tout ou partie du programme en mémoire	63
LOAD	Charge un programme à partir d'un périphérique	64
LOAD DOS	Charge en mémoire un programme	65
LOG	Donne le logarithme naturel d'un argument	67
MID$	Extrait des caractères d'une chaîne	68
"*“	Multiplie l'élément gauche par le droit	69
NEW	Supprime le programme en mémoire	70
NEW DOS	Format ou efface une disquette	72
NEXT	Renvoie à la ligne qui suit FOR	73
NOT	Inverse une expression	74
ON GOSUB	La valeur de ON appelle un sous-programme	75
ON GOTO	La valeur de ON appelle une ligne de programme	76
OPEN	Ouvre un fichier	77
OR	Opérateur de relation logique OU	78
()	Délimitent des opérations arithmétiques	80
PEEK	Donne la valeur d'une adresse	81
".“	Remplace la virgule dans un nombre décimale	82
";“	Sépare des chaînes ou des variables	83
POKE	Place une valeur à une adresse	84
POS	Donne la position du curseur	85
"%“	Variable entière	86

PRINT	Affiche une chaîne ou le contenu d'une variable	87
PRINT #	Ecrit sur un fichier	88
" ↑ "	Elève un nombre à une puissance	89
READ	Lit le contenu de DATA	90
REM	Insère un commentaire dans un programme	91
RENAME	Change le nom d'un fichier	92
RESTORE	Permet de relire des données	93
RETURN	Retour à la ligne qui suit le dernier GOSUB	95
RIGHT$	Extrait des caractères droits d'une chaîne	96
RND	Donne un nombre au hasard	97
RUN	Lance l'exécution d'un programme	98
SAVE	Enregistre un programme	100
SAVE DOS	Sauvegarde un programme	101
SCRATCH	Supprime un fichier	102
SGN	Donne le signe d'un argument	103
SIN	Donne le sinus d'un argument radians	105
" — "	Soustrait l'élément droit du gauche	106
SPC	Insère des espaces	107
SQR	Extrait la racine carrée	108
STEP	Donne le pas d'incrémentation	109
STOP	Provoque une interruption	110
STR$	Convertit une valeur en chaîne	111
" > "	Compare deux valeurs ou deux chaînes	112
" > ="	Compare deux valeurs ou deux chaînes	113
SYS	Exécute un programme en langage machine	114
TAB	Définit le début d'un affichage	115
TAN	Donne la tangente d'un argument radians	116
THEN	Conséquence d'une condition	117
TO	Fin de la variable de contrôle	118
USR	Place un argument dans un programme machine	120
VAL	Convertit une chaîne en valeur	121
VALIDATE	Réorganise la disquette	123
VERIFY	Compare le programme résidant à l'enregistrement	124
","	Sépare divers éléments	125
WAIT	Suspend le déroulement d'un programme	126

INTRODUCTION

« PASSEPORT POUR COMMODORE 64 » permet de se servir immédiatement du Basic. Un exemple de programme accompagne chaque fonction, instruction ou commande. Tous les mots clés sont répertoriés page par page dans l'ordre alphabétique, la recherche est donc aisée et rapide.

Si vous êtes débutant, vous pourrez apprendre facilement, en vous amusant, à entrer les programmes. Vous devrez le faire consciencieusement en cherchant à en comprendre chaque ligne ; une explication détaillée vous y aidera. Lorsque vous maîtriserez le Basic du COMMODORE 64, vous aurez toujours la possibilité de trouver dans ce manuel une syntaxe momentanément oubliée.

Si vous connaissez déjà le Basic, vous savez très bien que les ordinateurs ont des jeux d'instructions différents. Ce manuel vous permettra de trouver immédiatement une syntaxe propre au COMMODORE 64.

Chaque page est rédigée comme suit :

— En titre le mot clé.
— La traduction anglais/français.
— L'utilisation.
— Un exemple de programme.
— Une explication détaillée.

Bien qu'étant simples, les programmes contiennent les différentes formulations propres au COMMODORE 64. Les arguments sont rarement identiques, c'est pourquoi ils sont représentés sous leurs différentes formes.

Deux modes d'exécution des programmes sont disponibles :

— Le mode direct : l'ordinateur répond dès que vous frappez RETURN.

— Le mode différé : le programme est mis en mémoire avant que son exécution ne soit lancée par une commande explicite. Chaque ligne commence par un numéro et se termine par RETURN. Le programme se déroule dans l'ordre croissant de ses numéros de lignes. Un programme en mode différé peut être conservé sur cassette ou disquette, ce qui n'est pas possible avec un programme en mode direct.

Avec le Basic du COMMODORE 64, les espaces ne sont pas significatifs, sauf à l'intérieur des chaînes de caractères.

ABS

TRADUCTION

Abréviation de ABSOLUTE, absolue.

UTILISATION

ABS donne la valeur absolue d'un argument.

```
10 REM ABS                    RUN
20 LET A=45-50                -5
30 PRINT A                    5
40 PRINT ABS (A)              23
50 PRINT ABS (-23)            -5
60 PRINT ABS (-25)-30         5
70 PRINT ABS (25-30)
```

EXPLICATION

Ligne 20 Affecte le résultat de l'opération 45-50 à la variable A.

30 Affiche le contenu de la variable A (donc -5).

40 Affiche la valeur absolue du contenu de la variable A (donc 5).

50 L'argument peut être un nombre.

60 Donne la valeur absolue du premier nombre puis en soustrait le suivant.

70 Affiche la valeur absolue du résultat de l'opération entre parenthèses.

ADDITION "+"

UTILISATION

Opère la somme de deux éléments.

```
10 REM ADDITION "+"
20 LET X=3
30 LET Y=5
40 PRINT X+Y
50 LET A$="BON": LET B$="JOUR"
60 PRINT A$+B$
70 IF (X=3)+(Y=10) THEN PRINT
   "UNE RELATION EST VRAIE"
```

```
RUN
8
BONJOUR
UNE RELATION EST VRAIE
```

EXPLICATION

Ligne 20 Affecte le chiffre 3 à la variable X.

30 Affecte le chiffre 5 à la variable Y.

40 Affiche le résultat de l'addition des nombres contenus dans les variables X et Y.

50 Affecte les chaînes de caractères aux variables alphanumériques A$ et B$.

60 Affiche le contenu de la variable A$, puis celui de la variable B$; (nous avons une concaténation de chaînes).

70 Le signe plus remplace l'opérateur de relations logiques OR. Une des relations (X=3) est vraie, donc la chaîne de caractères entre guillemets s'affiche.

NOTA

Voir PARENTHESES, (priorités).

AFFECTATION "="

UTILISATION

Sépare les deux éléments d'une instruction, en affectant celui de droite à celui de gauche.

```
 10 REM AFFECTATION "="
 20 LET B=5
 30 LET C=B
 40 IF C=5 THEN PRINT "C=B"
 50 LET A$="AS"
 60 IF A$="AS" THEN PRINT "A$=AS"

RUN
C=B
A$=AS
```

EXPLICATION

Ligne 20 Affecte le chiffre 5 à la variable B.

30 Affecte la variable B à la variable C.

40 Si le contenu de la variable C est égal à 5, la chaîne "C=B" s'affiche.

50 Affecte la chaîne de caractères "AS" à la variable alphanumérique A$.

60 Si le contenu de la variable alphanumérique A$ est égal à "AS", la chaîne de caractères "A$=AS" s'affiche.

AND

TRADUCTION

AND, et.

UTILISATION

Dans une instruction IF...THEN, les deux comparaisons séparées par AND doivent être vraies pour que s'exécute la commande qui suit THEN.

```
 10 REM AND
 20 LET A$="LES 2 CONDITIONS SONT VRAIES"
 30 IF 2<3 AND 9>8 THEN PRINT A$
 40 IF "A"<"B" AND "D">"C" THEN PRINT A$
 50 IF (1<2) * (3>2) THEN PRINT A$
 60 PRINT 1<2 AND 3>2
 70 PRINT 1=2 AND 3>2

RUN
LES 2 CONDITIONS SONT VRAIES
LES 2 CONDITIONS SONT VRAIES
LES 2 CONDITIONS SONT VRAIES
-1
 0
```

EXPLICATION

Ligne 20 Affecte la chaîne de caractères à la variable A$.

Ligne 30 Si les deux relations sont vraies, le contenu de la variable A$ s'affiche.

40 Idem ligne 30; ici l'ordinateur se réfère à la valeur
du code de chaque lettre.
50 Le signe * remplace AND.
60 Les deux relations sont vraies, donc 1 s'affiche.
70 Une des deux relations est fausse, donc 0 s'affiche.

ASC

TRADUCTION

ASC, abréviation de ASCII.

UTILISATION

Donne le code ASCII.

```
 10 REM ASC
 20 PRINT ASC ("A")
 30 PRINT ASC ("AC")
 40 LET A$="B"
 50 PRINT ASC (A$)
 60 PRINT ASC ("A+B")

RUN
65
65
66
65
```

EXPLICATION

Ligne 20 Affiche le code ASCII de la lettre A.
30 Affiche le code ASCII de la première lettre.
40 Affecte la lettre B à la variable A$.
50 Affiche le code ASCII de la lettre contenue dans la variable A$.
60 Idem ligne 30.

ATN

TRADUCTION

Abréviation de ARC TANGENT, arc tangente.

UTILISATION

ATN calcule l'arc tangente d'un argument exprimé en radians.

```
10 REM ATN
20 PRINT ATN (2)
30 PRINT ATN (2)*4-5
40 LET B=ATN (2*4-5)
50 PRINT B
60 LET A=1
70 PRINT ATN (A)
80 PRINT ATN (A+2)
90 PRINT ATN (TAN (1.2490458))
```

```
RUN
1.10714872
-.57140513
1.24904577
 .785398163
1.24904577
1.2490458
```

EXPLICATION

Ligne 20 Affiche l'arc tangente de l'argument entre parenthèses.

30 Donne l'arc tangente du premier nombre puis multiplie le résultat par 4 et soustrait 5.

40 L'arc tangente de l'argument entre parenthèses est affecté à la variable B.

50 Affiche le contenu de la variable B.

60 Le nombre est affecté à la variable A.

70 Affiche l'arc tangente du contenu de la variable A.

```
80 L'argument entre parenthèses peut contenir des
   variables, des nombres et des opérations
   arithmétiques.
90 L'arc tangente et la tangente sont des fonctions
   mutuellement inverses.
```

CHR$

TRADUCTION

Abréviation de CHARACTER, caractères.

UTILISATION

CHR$ donne le caractère correspondant au code décimal de l'argument.

```
10 REM CHR$                    RUN
20 LET B=65                    A
30 PRINT CHR$ (B)              C
40 PRINT CHR$ (B+2)            W
50 PRINT CHR$ (39*2+9)         R
60 PRINT CHR$ (82)
```

EXPLICATION

Ligne 20 Affecte 65 à la variable B.

30 Affiche la lettre correspondant au code contenu dans la variable B.

40 Affiche la lettre correspondant au code contenu dans la variable B plus 2.

50 Affiche la lettre correspondant au code qui résulte de l'opération entre parenthèses.

60 Affiche la lettre correspondant au code du nombre 82.

CLOSE

TRADUCTION

CLOSE, fermer.

UTILISATION

Ferme un fichier.

```
10 REM CLOSE
20 OPEN 2,8,1,"EXEMPLE"
30 PRINT#2,"BONJOUR"
40 CLOSE 2
```

EXPLICATION

Ligne 20 Ouvre en écriture, le fichier 2 nommé EXEMPLE, sur le périphérique 8. Le premier argument est le n° de fichier qui sert de référence aux autres commandes. Le deuxième est le n° de périphérique. Le troisième indique l'écriture ou la lecture.

30 Ecrit dans le fichier n°2 la chaîne de caractères entre guillemets. Nous pouvons écrire des contenus de variables entières, numériques ou alphanumériques.

40 Ferme le fichier n°2. Lorsque le travail sur un fichier est terminé, ne jamais oublier de le fermer.

NOTA

Voir OPEN.

CLR

TRADUCTION

Abréviation de: CLEAR, transparent.

UTILISATION

CLR réinitialise, à zéro, les variables numériques et, à vide, les variables alphanumériques.

```
 10 REM CLR
 20 LET A=12
 30 LET A$="ABC"
 40 CLR
 50 PRINT A
 60 PRINT A$
 70 PRINT A

RUN
 0

 0
```

EXPLICATION

Ligne 20 Affecte 12 à la variable numérique A.
30 Affecte "ABC" à la variable alphanumérique A$.
40 Vide les variables.
50 La variable numérique A contient un 0.
60 La variable alphanumérique est vide.
70 Idem ligne 50.

CLR/HOME

TRADUCTION

Abréviation de CLEAR, effacer et HOME, origine.

UTILISATION

CLR/HOME efface l'écran et positionne le curseur en haut à gauche.

```
10 REM CLR/HOME
20 PRINT "CLR/HOME"
30 FOR I=1 TO 750: NEXT
40 PRINT SPC(11)"APPUYEZ SUR STOP"
50 FOR I=1 TO 500: NEXT
60 GOTO 20

RUN
APPUYEZ SUR STOP
```

EXPLICATION

Ligne 20 CLR/HOME est obtenu avec SHIFT. Sur le listing un coeur représente cette instruction.

30 Arrêt 7/10" de secondes.

40 Affiche à la 11" colonne la chaîne de caractères entre guillemets.

50 Arrêt 5/10" de secondes.

60 Saut à la ligne 20, (pour arrêter la boucle appuyez sur STOP).

CMD

UTILISATION

CMD choisit un périphérique en sortie.

```
OPEN 1,4
CMD 1
LIST
PRINT# 1
CLOSE 1
```

EXPLICATION

OPEN Ouvre le fichier 1 qui donne accès au périphérique 4 (imprimante).
CMD Envoie vers le fichier 1 la commande suivante.
LIST Ce listing ira vers l'imprimante.
PRINT Revient à l'écran.
CLOSE Tout fichier doit être fermé après son utilisation.

NOTA
Voir OPEN.

CONT

TRADUCTION

Abréviation de CONTINUE, continuer.

UTILISATION

CONT permet de poursuivre un programme arrêté par STOP.

```
10 REM CONT
20 PRINT "TAPEZ SUR CONT"
30 END
40 PRINT "C'EST ";
50 PRINT "LA ";
60 PRINT "FIN"
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères.

30 Stoppe le programme. Pour continuer, appuyez sur CONT.

40 Si vous avez tapé CONT, la chaîne s'affiche.

50 Idem ligne 40.

60 Idem ligne 40.

COPY DOS

TRADUCTION

COPY, copier.

UTILISATION

Copie un fichier.

```
OPEN 15,8,15
PRINT# 15,"COPY0:new=1:old"
P shift R 15,"C0:new=1:old"
CLOSE 15
```

EXPLICATION

Ligne 1 Ouvre le fichier 15 du périphérique 8, le dernier argument est le numéro de canal.

2 Le fichier old du drive 1 sera copié sur le drive 0 et se nommera new. Si vous omettez 1 et 0 old du drive 0 se copiera sur la même disquette sous le nom de new.

3 Idem ligne 2, mais en abrégé.

4 Ferme le fichier 15.

COS

TRADUCTION

Abréviation de COSINE, cosinus.

UTILISATION

COS donne le cosinus d'un argument exprimé en radians.

```
10 REM COS
20 PRINT COS (2)
30 PRINT COS (2)*4-5
40 LET B=COS (2*4-5)
50 PRINT B
60 LET A=1
70 PRINT COS (A)
80 PRINT COS (A+2)
```

```
RUN
-.416146836
-6.66458734
-.989992497
 .540302306
-.989992497
```

EXPLICATION

Ligne 20 Affiche le cosinus de l'argument entre parenthèses.

30 Donne le cosinus du premier nombre puis multiplie le résultat par 4 et soustrait 5.

40 Le cosinus de l'argument entre parenthèses est affecté à la variable B.

50 Affiche le contenu de la variable B.

60 Le nombre est affecté à la variable A.

70 Affiche le cosinus du contenu de la variable A.

80 L'argument entre parenthèses peut contenir des variables, des nombres et des opérations arithmétiques.

DATA

TRADUCTION

DATA, donnée

UTILISATION

DATA affecte des données à un tableau.

```
 10  REM  DATA
 20  INPUT "TAPEZ UN NOMBRE ENTRE UN ET SEPT :";J
 30  IF J > 7 OR J < 1 THEN  PRINT "MAUVAIS": GOTO  20
 40  FOR I = 1 TO J
 50  READ J$
 60  DATA LUNDI,MARDI,MERCREDI,JEUDI,VENDREDI,SAMEDI,DIMANCHE
 70  NEXT I
 80  PRINT J$
 90  RESTORE
100  GOTO 20
```

EXPLICATION

Ligne 20 Affectez un nombre à la variable J.

30 Si le nombre n'est pas dans le créneau 1 à 7, retour ligne 20.

40 Affecte 1 à J, à la variable de contrôle I.

50 Lecture des données du tableau J$.

60 Affecte les données au tableau J$, (10 maximum voir DIM).

70 Retour ligne 50 un nombre de fois équivalent au contenu de J.

80 Affiche la dernière donnée lue.
90 Permet de reprendre la lecture des données à leur début.
100 Le programme recommence (Pour arrêter appuyez sur STOP).

DEF FN

TRADUCTION

Abréviation de DEFINE, définir, et de FONCTION, fonction.

UTILISATION

DEF FN affecte une fonction à une variable pour son utilisation ultérieure.

```
10 REM DEF FN                    RUN
20 DEF FN A(Z) = 3 * Z + 2        77
30 PRINT FN A(25)                 24
40 DEF FN B(Y) = 3 * 4 + 12
50 LET G = FN B(123)
60 PRINT G
```

EXPLICATION

Ligne 20 Affecte à la variable A, la fonction 3*Z+2.

30 Affiche le contenu de A. Z prend la valeur de l'argument entre parenthèses (25). Essayez le programme en changeant la valeur de cet argument.

40 Définit la fonction B comme étant 3*4+12. Ici Y est fictif mais indispensable (il peut être remplacé par n'importe quoi).

50 Affecte la fonction B à la variable G. 123 est fictif mais indispensable (il peut être remplacé par n'importe quoi).

60 Affiche le contenu de G.

DIFFERENT "<>"

UTILISATION

Permet d'effectuer une comparaison entre deux valeurs numériques, deux expressions arithmétiques logiques, deux chaînes de caractères ou deux variables numériques ou alphanumériques.

```
10 REM DIFFERENT DE
20 LET A=12
30 LET B=13
40 IF A<>B THEN PRINT A;"<>";B
50 LET A$="BON"
60 LET B$="BIEN"
70 IF A$<>B$ THEN PRINT A$;"<>";B$
```

```
RUN
12<>13
BON<>BIEN
```

EXPLICATION

Ligne 20 Affecte le chiffre 12 à la variable numérique A.

30 Affecte le chiffre 13 à la variable numérique B.

40 Si le contenu de la variable A est différent de celui de la variable B, l'instruction qui suit THEN s'exécute.

50 Affecte la chaîne de caractères "BON" à la variable alphanumérique A$.

60 Affecte la chaîne de caractères "BIEN" à la variable alphanumérique B$.

70 Si le contenu de la variable A$ est différent de celui de la variable B$, l'instruction qui suit THEN s'exécute.

DIM

TRADUCTION

Abréviation de DIMENSION, dimension.

UTILISATION

Réserve la place nécessaire à un tableau.

```
10 REM DIM
20 DIM A(15)
30 FOR B = 1 TO 15
40 READ A(B)
50 PRINT A(B)
60 NEXT B
70 DATA 1,2,3,4,5,6,7,8,9,12,23,24,35,11,22
```

EXPLICATION

Ligne 20 Permettra de rentrer 15 données dans le tableau A.
30 Affecte 1 à 15, à la variable de contrôle B.
40 Lecture des données du tableau A (B est facultatif).
50 Ecrit les données du tableau A (B est facultatif).
60 Retour ligne 40 jusqu'à ce que le contenu de B soit au maximum.
70 Affecte les données au tableau A.

NOTA

En supprimant la ligne 20, vous n'afficherez que les 10 premières données du tableau, puis vous aurez un message d'erreur. Sans DIM l'ordinateur permet d'entrer seulement 10 données.

DIM peut avoir 3 arguments. ex: DIM A (6,3,2)

DIVISION .../..

UTILISATION

/ divise un élément par un autre.

```
10 REM DIVISION
20 LET A=18/3
30 PRINT A
40 LET B=2
50 PRINT A/B
60 LET C=A*5/B
70 PRINT C
```

```
RUN
6
3
15
```

EXPLICATION

Ligne 20 Affecte le résultat de l'opération, 18 divisé par 3, à la variable numérique A.

30 Affiche le contenu de la variable A.

40 Affecte le chiffre 2, à la variable numérique B.

50 Affiche le résultat du contenu de la variable A, divisé par celui de la variable B.

60 Affecte le résultat de l'opération, contenu de A multiplié par 5 et divisé par contenu de B, à la variable numérique C.

70 Affiche le contenu de C.

NOTA

Voir PARENTHESES (priorités).

DOLLAR "$"

UTILISATION

$ définit une variable comme étant du type alphanumérique.

```
 10 REM DOLLAR ($)
 20 LET A$="UN+TROIS": LET B$="=QUATRE"
 30 PRINT A$;B$
 40 PRINT CHR$ (65)
 50 LET C$=STR$ ( 54*11)
 60 PRINT C$
 70 PRINT LEN C$
RUN
UN+TROIS=QUATRE
A
 594
 4
```

EXPLICATION

Ligne 20 Affecte les chaînes de caractères aux variables alphanumériques A$ et B$.

30 Affiche sur une ligne le contenu des variables alphanumériques, A$ et B$.

40 Affiche la lettre correspondant au code 65.

50 Convertit l'opération entre parenthèses en chaîne de caractères, puis affecte cette chaîne à la variable alphanumérique C$.

60 Affiche le contenu de la variable alphanumérique C$.

70 Donne la longueur de la chaîne contenue dans la variable C$.

DUPLICATE DOS

TRADUCTION

DUPLICATE, faire une copie.

UTILISATION

Copie une disquette sur une autre.

```
OPEN 15,8,15
PRINT# 15,"DUPLICATE"
P shift R 15,"D"
CLOSE 15
```

EXPLICATION

Ligne 1 Ouvre le fichier 15 du périphérique 8, le dernier argument est le numéro de canal.

2 Copie entièrement une disquette sur une autre. Deux drives sont absolument nécessaires.

3 Idem ligne 2, mais en abrégé.

4 Ferme le fichier 15.

E

TRADUCTION

Abréviation de EXPONENTIAL, exponentiel.

UTILISATION

Lorsqu'un nombre a plus de huit zéros, E les transforme en exposant.

```
 10 REM (E)
 20 LET A=1
 30 FOR B=1 TO 40
 40 LET A=A*10
 50 PRINT A
 60 NEXT B

RUN
10
  V
  V
100000000
1E+9
  V
  V
1E+38
overflow error in 40
```

EXPLICATION

Ligne 20 Affecte le chiffre 1 à la variable numérique A.
     30 Affecte les chiffres 1 à 40, à la variable de contrôle B.

40 Affecte le contenu de la variable A multiplié par 10, à la variable A.
50 A chaque passage, le nouveau contenu de la variable A s'affiche.
60 Renvoie à la ligne 40 jusqu'à ce que B soit à sa valeur maximum.

NOTA
Vous pouvez avoir une notation négative, (ex -1E+38).
Essayez le programme en transformant la ligne 20 comme ceci:
20 LET A=-1

EGAL

UTILISATION

EGAL, employé avec IF, permet d'effectuer une comparaison.

```
 10 REM EGAL (=)
 20 INPUT "TAPEZ UN NOMBRE :";A
 30 INPUT "UN AUTRE :";B
 40 IF A=B THEN PRINT A;"=";B: GOTO 60
 50 PRINT A;"<>";B
 60 INPUT "TAPEZ UN MOT :";A$
 70 INPUT "UN AUTRE :";B$
 80 IF A$=B$ THEN PRINT A$;"=";B$: END
 90 PRINT A$;"<>";B$
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères entre guillemets et demande d'affecter un nombre à la variable A.

30 Idem ligne 20.

40 Si le contenu de A est égal au contenu de B saut à la ligne 60.

50 Si la condition de la ligne 40 n'est pas satisfaite, cette ligne affiche le contenu de A, la chaîne de caractères, puis le contenu de B.

60 Affiche la chaîne de caractères entre guillemets, puis demande d'affecter un mot à la variable A$.

70 Idem ligne 60.

80 Si le contenu de A$ est égal au contenu de B$, le programme s'arrête.

90 Si la condition de la ligne 80 n'est pas satisfaite, cette ligne affiche le contenu de A$, la chaîne de caractères, puis le contenu de B$.

END

TRADUCTION

END, arrêt.

UTILISATION

END arrête le déroulement d'un programme.

```
10 REM END
20 PRINT "UN"
30 END
40 PRINT "DEUX"
50 INPUT A$
60 IF A$="S" THEN END
70 PRINT "FIN"
80 END
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères entre guillemets.
30 Interrompt momentanément le déroulement du programme. (CONT permet de le continuer).
40 Affiche la chaîne de caractères entre guillemets.
50 Affectez un mot à la variable A$.
60 Si vous entrez S le programme s'arrête.
70 Affiche la chaîne de caractères entre guillemets.
80 END à la fin d'un programme est facultatif.

ERREUR

BAD DATA (mauvaise donnée). Un fichier nous a donné une chaîne de caractères, alors que nous attendions une valeur numérique.

BAD SUBSCRIPT (mauvais indice). Mauvaise dimension de tableau.

CAN'T CONTINUE (impossible de continuer). La commande CONT refuse de fonctionner.

DEVICE NOT PRESENT (périphérique absent). Le périphérique voulu par OPEN, CLOSE, CMD, PRINT, INPUT, ou GET n'est pas connecté.

DIVISION PAR ZERO (division par zéro). Il est impossible de diviser par zéro.

EXTRA IGNORED (supplément ignoré). On a tapé trop d'éléments en réponse à un INPUT.

FILE NOT FOUND (fichier non trouvé). Sur bande le fichier a une marque de fin, sur disque il n'existe pas de fichier de ce nom.

FILE NOT OPEN (fichier non ouvert). Le fichier spécifié dans CMD, PRINT, INPUT, ou GET doit d'abord être ouvert par OPEN.

FILE OPEN (fichier déjà ouvert). Il est impossible d'ouvrir un fichier avec un numéro déjà utilisé.

FORMULA TOO COMPLEX (formule trop complexe). L'expression en cours d'évaluation doit être fractionnée.

ILLEGAL DIRECT (mode direct illégal). Les instructions INPUT et DEF FN ne peuvent être utilisées en mode direct.

ILLEGAL QUANTITY (quantité illégale). L'argument est hors intervalle.

LOAD (chargement). Problème lors du chargement d'un programme sur bande.

NEXT WITHOUT FOR (NEXT sans FOR). Mauvaise imbrication de boucles, ou variable de NEXT ne correspondant pas à celle de FOR.

NOT INPUT FILE (ce fichier n'est pas en entrée). Tentative d'INPUT ou de GET sur un fichier spécifié en écriture.

NOT OUTPUT FILE (ce fichier n'est pas en sortie). Tentative de PRINT sur un fichier spécifié en lecture.

OUT OF DATA (données épuisées). On essaie d'effectuer un READ alors qu'il ne reste plus de données dans les instructions DATA.

OUT OF MEMORY (mémoire pleine). Il ne reste plus de mémoire pour le programme ou les données. Cela peut aussi se produire si on a trop de boucles ou de sous-programmes.

OVERFLOW (débordement). Le résultat du calcul dépasse 1,70141884e+38.

REDIM'D ARRAY (tableau redimensionné). Un tableau ne peut pas être dimensionné plusieurs fois.

REDO FROM START (recommencez au début). INPUT attendait des données numériques, mais a reçu des données alphabétiques.

RETURN WITHOUT GOSUB (RETURN sans GOSUB). On a rencontré RETURN sans être dans un sous-programme.

STRING TOO LONG (chaîne trop longue). Une chaîne ne doit pas dépasser 255 caractères.

SYNTAX (syntaxe). Erreur de syntaxe qui peut être due à une mauvaise parenthèse, ou à un mot clé mal orthographié.

TYPE MISMATCH (désaccord de type). On a utilisé une valeur numérique à la place d'une chaîne de caractères, ou vice versa.

UNDEF'D FUNCTION (fonction non définie). Une fonction doit être définie par DEF FN.

UNDEF'D STATEMENT (instruction non définie). Tentative de branchement par GOTO, GOSUB, ou RUN à un numéro qui n'existe pas.

VERYFY (vérification). Le programme sur bande ou disque ne concorde pas avec celui en mémoire.

EXP

TRADUCTION

Abréviation de EXPONENTIAL, exponentielle.

UTILISATION

EXP élève la constante E (2.7182818) à la puissance de l'argument.

```
10 REM EXP
20 LET A=1
30 PRINT EXP (A)
40 PRINT EXP (A)+1
50 LET B=EXP (3)
60 PRINT B
70 PRINT EXP (3*2+4-3)
```

```
RUN
2.71828183
3.71828183
20.0855369
1096.63316
```

EXPLICATION

Ligne 20 Affecte le chiffre 1 à la variable A.

30 La constante E est élevée à la puissance du contenu de la variable A.

40 La constante E est élevée à la puissance du contenu de la variable A, puis 1 est additionné à ce résultat.

50 La constante E élevée à la puissance 3, est affectée à la variable B.

60 Affiche le contenu de la variable B.

70 La constante E est élevée à la puissance de l'argument entre parenthèses.

FN

TRADUCTION

Abréviation de FONCTION, fonction.

UTILISATION

Donne la valeur d'une fonction définie par DEF FN.

```
10 REM FN                          RUN
20 DEF FN A(Z) = 3 * Z + 2         77
30 PRINT FN A(25)                  24
40 DEF FN B(Y) = 3 * 4 + 12
50 LET G = FN B(123)
60 PRINT G
```

EXPLICATION

Ligne 20 Affecte à la variable A, la fonction 3*Z+2.

30 Affiche le contenu de A. Z prend la valeur de l'argument entre parenthèses (25). Essayez le programme en changeant la valeur de cet argument.

40 Définit la fonction B comme étant 3*4+12. Ici Y est fictif mais indispensable (il peut être remplacé par n'importe quoi).

50 Affecte la fonction B à la variable G. 123 est fictif mais indispensable (il peut être remplacé par n'importe quoi).

60 Affiche le contenu de G.

FOR

TRADUCTION

FOR, pour.

UTILISATION

Affecte des nombres à une variable de contrôle.

```
 10 REM FOR
 20 FOR A=1 TO 9
 30 PRINT A
 40 NEXT A
 50 PRINT "L'ECRAN VA S'EFFACER"
 60 FOR I=1 TO 1000: NEXT I
 70 PRINT "CLR/HOME"
 80 LET B=10: LET C=15
 90 FOR X=B TO C
100 PRINT X
110 NEXT X
```

EXPLICATION

Ligne 20 Affecte 1 à 9, à la variable de contrôle A.
30 Affiche à chaque passage le nouveau contenu de A.
40 Renvoie ligne 30 jusqu'à ce que la variable A soit à sa valeur maximum.
50 Affiche la chaîne de caractères entre guillemets.
60 Reste ligne 60, jusqu'à ce que la variable I soit à sa valeur maximum (boucle d'attente d'une seconde).
70 Efface l'écran.
80 Affecte les nombres 10 et 15 aux variables B et C.

```
 90 Des variables numériques peuvent être utilisées.
100 A chaque passage, le nouveau contenu de X s'affiche.
110 Renvoie ligne 100 jusqu'à ce que la variable X soit
    à sa valeur maximum.
```

FRE

TRADUCTION

Abréviation de FREE, libre.

UTILISATION

FRE donne le nombre d'octets disponible.

PRINT FRE(1) Vous affiche le nombre d'octets inoccupés dans la mémoire. Si ce nombre est supérieur à 32K, vous aurez une réponse négative. Pour avoir le nombre exact, ajoutez 65536 ou faites PRINT FRE(1)+65536. L'argument est aléatoire mais nécessaire.
FRE peut être employé dans un programme.
Exemple:

```
10 PRINT FRE(255)
20 X=FRE(3)
30 PRINT X
```

GET

TRADUCTION

GET, prendre.

UTILISATION

Affecte à une variable le caractère tapé.

```
10 REM GET
20 PRINT "CLR/HOME"
30 GET A$
40 PRINT "POUR ARRETER TAPER: S"
50 IF A$ = "B" THEN PRINT "BON"
60 IF A$ = "M" THEN PRINT "MAUVAIS"
70 IF A$ = "S" THEN STOP
80 FOR I = 1 TO 500: NEXT
90 GOTO 20
```

EXPLICATION

Ligne 20 Efface l'écran.

30 Lorsque le programme passe sur cette ligne, la touche enfoncée affecte son caractère à la variable A$.

40 Affiche la chaîne de caractères entre guillemets.

50 Si vous avez tapé B, BON s'affiche.

60 Si vous avez tapé M, MAUVAIS s'affiche.

70 Si vous avez tapé S, le programme s'arrête.

80 Boucle d'attente 1/2 seconde.

90 Ce retour à la ligne 20 permet de passer sur la ligne 30 pendant qu'une touche est enfoncée.

GET

TRADUCTION

GET#, prendre.

UTILISATION

Prend lettre par lettre des données sur fichier.

```
10 REM GET#
20 OPEN 2,8,0,"EXEMPLE"
30 GET#2,A$
40 CLOSE 2
50 PRINT A$
```

EXPLICATION

Ligne 20 Ouvre en écriture, le fichier 2 nommé EXEMPLE, sur le périphérique 8. Le premier argument est le n° de fichier qui sert de référence aux autres commandes. Le deuxième est le n° de périphérique. Le troisième indique l'écriture ou la lecture.

30 Affecte la première lettre du fichier n°2 à la variable alphanumérique A$. Nous pouvons avoir des variables entières, numériques ou indicées.

40 Ferme le fichier n°2. Lorsque le travail sur un fichier est terminé ne jamais oublier de le fermer.

50 Affiche sur l'écran le contenu de A$.

NOTA

Voir OPEN.

GOSUB

TRADUCTION

GO, aller et SUB, sous.

UTILISATION

GOSUB envoie à un sous-programme.

```
 10 REM GOSUB                    RUN
 20 PRINT "LIGNE 20"             LIGNE 20
 30 GOSUB 80                     SOUS PROGRAMME
 40 PRINT "LIGNE 40"             LIGNE 40
 50 GOSUB 80                     SOUS PROGRAMME
 60 PRINT "LIGNE 60"             LIGNE 60
 70 END
 80 PRINT "SOUS PROGRAMME"
 90 RETURN
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères entre guillemets.
30 Envoie au sous-programme débutant à la ligne 80.
40 Affiche la chaîne de caractères entre guillemets.
50 Envoie au sous-programme débutant à la ligne 80.
60 Affiche la chaîne de caractères entre guillemets.
70 Stoppe le programme.
80 Affiche la chaîne de caractères entre guillemets.
90 Renvoie à la ligne qui suit le dernier GOSUB rencontré.

NOTA

Un sous-programme peut contenir plusieurs lignes.

GOTO

TRADUCTION

GO, aller et TO, à.

UTILISATION

GOTO fait un saut à une ligne spécifiée.

```
10 REM GOTO
20 PRINT "LIGNE 20"
30 GOTO 70
40 PRINT "LIGNE 40"
50 GOTO 90
60 PRINT "LIGNE 60"
70 PRINT "LIGNE 70"
80 GOTO 40
90 PRINT "FIN"
```

```
RUN
LIGNE 20
LIGNE 70
LIGNE 40
FIN
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères entre guillemets.
30 Envoie à la ligne 70.
40 Affiche la chaîne de caractères entre guillemets.
50 Envoie à la ligne 90.
60 Cette ligne est ignorée.
70 Affiche la chaîne de caractères entre guillemets.
80 Renvoie à la ligne 40.
90 Affiche la chaîne de caractères entre guillemets.

GUILLEMETS " "

UTILISATION

Les guillemets servent à délimiter une chaîne de caractères.

```
10 REM GUILLEMETS (" ")
20 PRINT "12 ";
30 PRINT "MOTS DE 11 LETTRES"
40 LET A=34
50 PRINT A;"+";2;"=";A+2
60 PRINT "LE CODE DE A EGALE ";
70 PRINT ASC ("AS")

RUN
12 MOTS DE 11 LETTRES
34+2=36
LE CODE DE A EGALE 65
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères, qui doit être obligatoirement entre guillemets. Le point-virgule empêche le retour chariot, l'affichage suivant se fera donc sur cette ligne.

30 Affiche la chaîne de caractères entre guillemets.

40 Affecte le nombre 34 à la variable numérique A.

50 Affiche sur une ligne, le contenu de A, la chaîne de caractères "+", le chiffre 2, la chaîne de caractères "=", puis le contenu de A plus 2.

60 Idem ligne 20.

70 Affiche le code de la première lettre de la chaîne de caractères "AS".

IF

TRADUCTION

IF, si.

UTILISATION

IF sert à tester une condition. Lorsque celle-ci est satisfaite, l'instruction qui suit THEN s'exécute.

```
 10 REM IF
 20 INPUT "TAPEZ UN NOMBRE :";A
 30 INPUT "UN AUTRE :";B
 40 IF A>B THEN PRINT A;">";B
 50 IF A<B THEN PRINT A;"<";B
 60 IF A=B THEN PRINT A;"=";B
 70 LET C$=" EST AVANT "
 80 INPUT "TAPEZ UNE LETTRE :";A$
 90 INPUT "UNE AUTRE :";B$
100 IF A$<B$ THEN PRINT A$;C$;B$
110 IF A$>B$ THEN PRINT B$;C$;A$
120 IF A$=B$ THEN PRINT "VOUS AVEZ TAPE DEUX FOIS ";A$
```

EXPLICATION

Ligne 20 Affectez un nombre à la variable numérique A.
30 Idem ligne 20.
40 Si le contenu de A est supérieur à celui de B, l'instruction qui suit THEN s'effectue.
50 Si A est inférieur à B, la chaîne s'affiche.
60 Si A est égal à B, la chaîne s'affiche.
70 Affecte la chaîne de caractères à C$.

```
 80 Affectez une lettre à la variable alphanumérique A$.
 90 Idem ligne 80.
100 Idem ligne 50.
110 Idem ligne 40.
120 Idem ligne 60.
```

INFERIEUR "<"

UTILISATION

Permet de comparer deux valeurs numériques ou deux chaînes de caractères.

```
 10 REM INFERIEUR
 20 LET A=4
 30 IF A<5 THEN PRINT "4 EST INFERIEUR A 5"
 40 LET A$="G"
 50 IF "D"<A$ THEN PRINT "D EST AVANT G"

RUN
4 EST INFERIEUR A 5
D EST AVANT G
```

EXPLICATION

Ligne 20 Affecte le chiffre 4 à la variable numérique A.

30 Si le contenu de la variable numérique A est inférieur à 5, la chaîne de caractères entre guillemets s'affiche.

40 Affecte la lettre "G" à la variable alphanumérique A$.

50 Si le code de la lettre "D", est inférieur au code de la lettre contenue dans la variable A$, la chaîne de caractères s'affiche.

INF. OU EGAL "<="

UTILISATION

Compare deux valeurs numériques ou deux chaînes de caractères.

```
10 REM INFERIEUR OU EGAL
20 LET A=4
30 IF A<=5 THEN PRINT "4 < 5"
40 IF A<=4 THEN PRINT "A EGALE 4"
50 LET A$="G"
60 IF "D"<=A$ THEN PRINT "D AVANT G"
70 IF "G"<=A$ THEN PRINT "A$ EGALE G"
```

```
RUN
4 < 5
A EGALE 4
D AVANT G
A$ EGALE G
```

EXPLICATION

Ligne 20 Affecte le chiffre 4 à la variable numérique A.

30 Si le contenu de A est inférieur ou égal à 5, la chaîne de caractères s'affiche.

40 Si le contenu de A est inférieur ou égal à 4, la chaîne de caractères s'affiche.

50 Affecte la lettre "G" à la variable A$.

60 Si le code de la lettre "D", est inférieur ou égal au code de la lettre contenue dans la variable A$, la chaîne de caractères s'affiche.

70 Si le code de la lettre "G", est inférieur ou égal au code de la lettre contenue dans la variable A$, la chaîne de caractères s'affiche.

INITIALIZE DOS

TRADUCTION

INITIALIZE, initialiser.

UTILISATION

Remet le drive à l'état initial.

```
OPEN 15,8,15
PRINT# 15,"INITIALIZE"
P shift R 15,"I"
CLOSE 15
```

EXPLICATION

Ligne 1 Ouvre le fichier 15 du périphérique 8, le dernier argument est le numéro de canal.

2 Après une mauvaise manipulation la lampe rouge du drive clignote, cette commande remet le drive à l'état initial (comme à la mise sous tension).

3 Idem ligne 2, mais en abrégé.

4 Ferme le fichier 15.

INPUT

TRADUCTION

INPUT, données.

UTILISATION

INPUT affecte des variables.

```
10 REM INPUT
20 PRINT "TAPEZ UN CHIFFRE"
30 INPUT A
40 INPUT "UN AUTRE :";B
50 PRINT "TAPEZ UN MOT"
60 INPUT A$
70 INPUT "UN AUTRE :";B$
80 PRINT "VOUS AVEZ TAPE";A;B;A$;" ET ";B$
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères entre guillemets.

30 Affectez un chiffre à la variable numérique A.

40 Affiche la chaîne de caractères entre guillemets. Puis vous demande d'affecter un chiffre à la variable numérique B.

50 Affiche la chaîne de caractères entre guillemets.

60 Affectez un mot à la variable alphanumérique A$.

70 Affiche la chaîne de caractères entre guillemets. Puis vous demande d'affecter un mot à la variable alphanumérique B$.

80 Affiche sur une ligne: la chaîne entre guillemets, le contenu de A, de B, de A$, "ET", puis le contenu de B$.

INPUT

TRADUCTION

INPUT#, entrer.

UTILISATION

Affecte des données à partir d'un périphérique.

```
10 REM INPUT#
20 OPEN 2,8,0,"EXEMPLE"
30 INPUT#2,A$
40 CLOSE 2
50 PRINT A$
```

EXPLICATION

Ligne 20 Ouvre en lecture, le fichier 2 nommé EXEMPLE, sur le périphérique 8. Le premier argument est le n° de fichier qui sert de référence aux autres commandes. Le deuxième est le n° de périphérique. Le troisième indique l'écriture ou la lecture.

30 Affecte les données du fichier n°2 à la variable alphanumérique A$. Nous pouvons avoir des variables entières, numériques ou indicées.

40 Ferme le fichier n°2. Lorsque le travail sur un fichier est terminé ne jamais oublier de le fermer.

50 Affiche sur l'écran le contenu de A$.

NOTA

Voir OPEN.

INT

TRADUCTION

Abréviation de INTEGER, nombre entier.

UTILISATION

INT extrait le plus grand nombre entier inférieur à l'argument.

```
10 REM INT
20 LET A=12.345
30 PRINT INT (A)
40 PRINT INT (A+2.5)
50 PRINT INT (A)+2.5
60 PRINT INT (3.75)
70 PRINT INT (-5.75)
80 PRINT INT ((9*4+56-5)/2)
```

```
RUN
12
14
14.5
3
-6
43
```

EXPLICATION

Ligne 20 Affecte le nombre 12,345 à la variable A.

30 Affiche l'entier du nombre contenu dans la variable A.

40 Affiche l'entier résultant de l'opération entre parenthèses.

50 Donne l'entier du contenu de A, puis additionne 2,5.

60 Affiche l'entier du nombre 3,75.

70 Le plus grand nombre entier inférieur à -5,75 est égal à -6.

80 L'argument entre parenthèses peut contenir toutes sortes d'opérations.

LEFT$

TRADUCTION

LEFT, gauche.

UTILISATION

Extrait un certain nombre de caractères gauches d'une chaine.

```
 10 REM LEFT$
 20 X$ = "C'EST FINI"
 30 PRINT LEFT$ (X$,6);"BIEN"; RIGHT$ (X$,5)
 40 PRINT LEFT$ (X$,(3 * 5) / 2);"ADE"
 50 PRINT LEFT$ ("CHANTEUR",5)

RUN
C'EST BIEN FINI
C'EST FADE
CHANT
```

EXPLICATION

Ligne 20 Affecte la chaine de caractères à la variable X$.

30 Affiche: les 6 premiers caractères de X$, le mot BIEN, puis les 5 derniers caractères de X$.

40 Affiche: les 7 premiers caractères de X$, puis le mot ADE. Lorsque l'argument contient des décimales il n'est retenu que l'entier.

50 Affiche les 5 premiers caractères du mot CHANTEUR.

LEN

TRADUCTION

Abréviation de LENGTH, longueur.

UTILISATION

LEN calcule la longueur d'une chaîne.

```
10 REM LEN                              RUN
20 LET A$="COMMODORE": LET B=64         COMMODORE 64 COMMODORE
30 PRINT A$;B;A$                        9
40 PRINT LEN (A$)                       3
50 PRINT LEN (STR$(B))                  12
60 PRINT LEN (A$+STR$(B))               6
70 PRINT LEN ("CHAINE")                 13
80 PRINT LEN ("CHAINE"+"COLLIER")
```

EXPLICATION

Ligne 20 Affecte la chaîne à A$ et 64 à B.

30 Affiche le contenu des variables A$, B et A$.

40 Donne la longueur de la chaîne contenue dans A$.

50 Donne la longueur de la chaîne contenue dans B (un espace est compté). Avec STR$ une variable numérique devient alphanumérique.

60 Donne la longueur totale des chaînes contenues dans les variables A$ et B.

70 Donne la longueur de la chaîne entre guillemets.

80 Donne la longueur totale des chaînes entre guillemets.

LET

TRADUCTION

LET, soit.

UTILISATION

LET affecte une chaine ou une valeur à une variable.

```
 10 REM LET
 20 LET A=10
 30 LET B=5*3
 40 LET C=A+3
 50 LET D=A*B
 60 PRINT A,B,C,D
 70 LET A$="BON"
 80 LET B$="JOUR"
 90 LET C$=A$+"HOMME"
100 LET D$=A$+B$
110 PRINT A$,B$,C$,D$

RUN
 10      15       13            150
BON     JOUR     BONHOMME      BONJOUR
```

EXPLICATION

Ligne 20 Affecte le nombre 10 à la variable numérique A.
30 Affecte le résultat de l'opération 5*3 à B.
40 Affecte le contenu de A plus 3 à C.
50 Affecte le contenu de A multplié par le contenu de B, à la variable numérique D.

```
 60 Affiche le contenu des variables A, B, C, D.
 70 Affecte "BON" à la variable A$.
 80 Affecte "JOUR" à la variable B$.
 90 Le contenu de A$ et la chaîne de caractères
    "HOMME", sont affectés à la variable alphanumérique
    C$.
100 Les contenus de A$ et B$, sont affectés à D$.
110 Affiche le contenu des variables A$, B$, C$ et D$.
```

LIST

TRADUCTION

LIST, liste.

UTILISATION

LIST affiche tout ou partie du programme en mémoire.

```
10 REM LIST
20 PRINT "CE PROGRAMME ";
30 PRINT "VOUS AFFICHE ";
40 PRINT "LE LISTING"
50 LIST
60 PRINT "CETTE LIGNE EST IGNOREE"
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères entre guillemets, l'affichage suivant sera sur cette ligne.
30 Idem ligne 20
40 Affiche la chaîne de caractères entre guillemets.
50 Donne un listing complet de ce programme.
60 Cette ligne est ignorée car après la commande LIST le programme s'arrête.

NOTA

Nous pouvons afficher une partie de programme. Ex:

LIST 30	Affiche la ligne 30.
LIST 30-	Affiche de la ligne 30 à la fin.
LIST-30	Affiche du début à la ligne 30.
LIST 30-50.	Affiche de la ligne 30 à la ligne 50.

LOAD

TRADUCTION

LOAD, charger.

UTILISATION

LOAD charge un programme, à partir d'un lecteur de cassettes ou de disquettes.

```
10 REM LOAD
20 REM VOIR AUPARAVANT SAVE
30 PRINT "DANS TROIS SECONDES"
40 PRINT "CE PROGRAMME"
50 PRINT "CHERCHERA SUR CASSETTE"
60 PRINT "LE PROGRAMME EXEMPLE"
70 FOR I=1 TO 3000: NEXT
80 LOAD "EXEMPLE"
```

EXPLICATION

Ligne 20 Effectuez auparavant le programme SAVE.
30 Affiche la chaîne de caractères entre guillemets.
40 Affiche la chaîne de caractères entre guillemets.
50 Affiche la chaîne de caractères entre guillemets.
60 Affiche la chaîne de caractères entre guillemets.
70 Immobilise le programme pendant trois secondes.
80 Charge en mémoire à partir d'une cassette le programme EXEMPLE.

NOTA

LOAD efface l'ancien programme résidant en mémoire. LOAD "EXEMPLE",8 charge le programme "EXEMPLE" à partir d'une unité de disquettes.

LOAD DOS

UTILISATION

Charge en mémoire, un programme.

```
LOAD"name",8
LOAD"$",8
LOAD"*",8
LOAD"BI*",8
```

EXPLICATION

Ligne 1 A partir du périphérique 8, (la disquette) le programme name se charge.

2 Charge le catalogue, attention il se met en mémoire comme un programme ordinaire. Pour le visualiser, faire LIST.

3 Charge le dernier programme utilisé, ou le premier du catalogue.

4 Charge le premier programme du catalogue commençant par BI. Le signe * remplace les dernieres lettres du nom.

NOTA

Sur la disquette maître se trouve le programme C-64, qui aide beaucoup la commande LOAD.

```
LOAD"C-*",8
RUN
```

Vous avez maintenant les commandes suivantes:

```
/name = Idem ligne 1.
/$    = Idem ligne 2.
```

/* = Idem ligne 3.
/CO* = Idem ligne 4.
)$ = Le catalogue (sans le mettre en mémoire).
) = Donne le type d'erreur, (lorsque la lampe rouge du DOS
clignote)
? peut remplacer)

LOG

TRADUCTION

Abréviation de LOGARITHM, logarithme et de NATURAL, naturel.

UTILISATION

LOG donne le logarithme naturel d'un argument.

```
10 REM LOG                  RUN
20 LET A=2.7182818           .99999999
30 PRINT LOG (A)            1.99999999
40 PRINT LOG (A)+1          3
50 LET B=LOG (20.085537)    2
60 PRINT B
70 PRINT LOG (EXP (2))
```

EXPLICATION

Ligne 20 Affecte le nombre à la variable A.

30 Affiche le logarithme naturel du contenu de la variable A.

40 Donne le logarithme naturel du contenu de A, puis 1 est additionné à ce résultat.

50 Le logarithme naturel du nombre, est affecté à la variable B.

60 Affiche le contenu de la variable B.

70 LOG et EXP sont des fonctions mutuellement inverses.

MID$

TRADUCTION

Abréviation de MIDDLE, milieu.

UTILISATION

Extrait un certain nombre de caractères d'une chaîne.

```
 10 REM MID$
 20 A$ = "ANTICONSTITUTIONNELLEMENT"
 30 PRINT MID$ (A$,5,12)
 40 A = 3
 50 PRINT MID$ ("IMPROPRE",A,6 / 2)
 60 PRINT MID$ ("AFFIRME",3)

RUN
CONSTITUTION
PRO
FIRME
```

EXPLICATION

Ligne 20 Affecte la chaîne de caractères à la variable A$.
30 A partir de la position 5 de la chaîne contenue dans A$, 12 caractères s'affichent.
40 Affecte le chiffre 3 à la variable numérique A.
50 L'argument peut contenir des opérations arithmétiques et des variables numériques.
60 Affiche tous les caractères à partir du 3e.

MULTIPLICATION "*"

UTILISATION

Multiplie l'élément de gauche par celui de droite.
* est employé pour éviter la confusion avec la lettre X.

```
10 REM MULTIPLICATION (*)
20 PRINT 12*5
30 LET A=6*2
40 LET B=8*3
50 LET C=A*B
60 PRINT A,B,C
70 REM * PEUT REMPLACER AND
80 IF (A=6*2)*(B=8*3)
THEN PRINT "LES DEUX RELATIONS SONT VRAIES"
```

```
RUN
 60
 12            24            288
LES DEUX RELATIONS SONT VRAIES
```

EXPLICATION

Ligne 20 Affiche le résultat de la multiplication 12*5.

30 Affecte le résultat de la multiplication 6*2, à la variable numérique A.

40 Affecte le résultat de 8*3 à B.

50 Le résultat de la multiplication, contenu de A par contenu de B, est affecté à la variable numérique C.

60 Affiche le contenu des variables A, B, C. Les virgules donnent une tabulation de 10 colonnes.

80 Le signe * remplace l'opérateur de relations logiques AND. Les deux relations doivent être vraies, pour que s'exécute l'instruction qui suit THEN.

NEW

TRADUCTION

NEW, nouveau.

UTILISATION

NEW supprime le programme en mémoire.

```
 10 REM NEW
 20 PRINT "CLR/HOME"
 30 PRINT ,"VOULEZ VOUS SUPPRIMER"
 40 PRINT SPC(14)"CE PROGRAMME?"
 50 PRINT SPC(250)"**********"
 60 PRINT TAB(10)"* TAPEZ OUI OU NON *"
 70 PRINT TAB(10)"**********"
 80 INPUT A$
 90 IF A$="OUI" THEN NEW
100 PRINT "CLR/HOME"
110 LIST
```

EXPLICATION

Ligne 20 Efface l'écran.
30 La chaîne s'affiche à la onzième colonne.
40 La chaîne s'affiche à la quinzième colonne.
50 250/40=6 reste 10, l'affichage suivant se fera donc sur la septième ligne et onzième colonne.
60 La chaîne s'affiche à la onzième colonne.
70 Idem ligne 60.
80 Affectez "OUI" ou "NON" à la variable A$.

```
 90 Si vous avez tapé "OUI", la mémoire se vide.
100 Efface l'écran.
110 Si NEW n'a pas été exécutée, un listing complet
    s'affiche.
```

NEW DOS

TRADUCTION

NEW, nouveau

UTILISATION

Formate ou efface une disquette.

```
OPEN 15,8,15
PRINT# 15,"NEW:name,id"
P shift R 15,"N:name,id"
PRINT# 15,"NEW:name"
CLOSE 15
```

EXPLICATION

Ligne 1 Ouvre le fichier 15 du périphérique 8, le dernier argument est le numéro de canal.

2 Lorsqu'une disquette est neuve, elle doit être formatée au type d'appareil qui l'utilisera. Le nom de la disquette sera name, id est nécessaire et sera présent sur chaque bloc.

3 Idem ligne 2, mais en abrégé.

4 Efface entièrement une disquette, ce qui est beaucoup plus rapide que de la reformater, name peut être changé.

5 Ferme le fichier 15.

NOTA

Il faut environ une minute et demie pour formater une disquette.

NEXT

TRADUCTION

NEXT, prochain.

UTILISATION

NEXT renvoie à la ligne qui suit l'instruction FOR.

```
 10 REM NEXT
 20 FOR A=1 TO 4
 30 PRINT A;
 40 NEXT
 50 PRINT
 60 FOR B=1 TO 3
 70 FOR C=6 TO 9
 80 PRINT C;
 90 NEXT C
100 PRINT
110 NEXT B
```

```
RUN
1 2 3 4
6 7 8 9
6 7 8 9
6 7 8 9
```

EXPLICATION

Ligne 20 Affecte 1 à 4, à la variable de contrôle A.
30 A chaque passage, la nouvelle valeur de A s'affiche.
Le point-virgule permet l'affichage sur une ligne.
40 Retour ligne 30, jusqu'à ce que A égale 4.
50 Permet au prochain affichage de changer de ligne.
60 Affecte 1 à 3, à la variable de contrôle B.
70 Affecte 6 à 9, à la variable de contrôle C.
80 A chaque passage, la nouvelle valeur de C s'affiche.
90 Retour ligne 80, jusqu'à ce que C égale 9.
100 Permet au prochain affichage de changer de ligne.
110 Retour ligne 70, jusqu'à ce que B égale 3.

NOT

TRADUCTION

NOT, ne...pas.

UTILISATION

NOT inverse une expression.

```
 10 REM NOT
 20 LET A$="9 EST SUPERIEUR A 6"
 30 LET A=9
 40 IF A>6 THEN PRINT A$
 50 IF NOT 12=3 THEN PRINT "UNE RELATION FAUSSE DEVIENT VRAIE"
 60 PRINT NOT 8<6
 70 PRINT NOT 8<>6
RUN
9 EST SUPERIEUR A 6
UNE RELATION FAUSSE DEVIENT VRAIE
-1
 0
```

EXPLICATION

Ligne 20 Affecte la chaîne de caractères à la variable A$.

30 Affecte la valeur 9 à la variable numérique A.

40 Si le contenu de A est supérieur à 6, le contenu de A$ s'affiche.

50 Par NOT 12=3 devient vrai, donc l'instruction qui suit THEN s'exécute.

60 Avec NOT 8 plus petit que 6 devient vrai, donc 1 s'affiche.

70 Avec NOT 8 différent de 6 devient faux, donc 0 s'affiche.

ON GOSUB

TRADUCTION

ON, sur; GO, aller et SUB, sous.

UTILISATION

Suivant la valeur qui suit ON, il y a saut à un sous programme.

```
10 REM ON...GOSUB
20 INPUT "TAPEZ 1 OU 2 :";A
30 ON A GOSUB 50,70
40 END
50 PRINT "VOUS AVEZ TAPE 1"
60 RETURN
70 PRINT "VOUS AVEZ TAPE 2"
80 RETURN
```

EXPLICATION

Ligne 20 Affectez 1 ou 2 à la variable A.

30 Suivant la valeur de A, vous avez un saut à la ligne 50 ou 70.

40 Si vous avez entré un chiffre qui ne correspond pas à la quantité d'arguments de GOSUB, cette ligne est immédiatement exécutée et arrête le programme.

50 Premier sous-programme.

60 Retour à la ligne qui suit le dernier GOSUB.

70 Deuxième sous-programme.

80 Retour à la ligne qui suit le dernier GOSUB.

ON GOTO

TRADUCTION

ON, sur; GO, aller et TO, à.

UTILISATION

Suivant la valeur qui suit ON, il y a saut à une ligne.

```
10 REM ON...GOTO
20 INPUT "TAPEZ 1 OU 2 :";A
30 ON A GOTO 50,70
40 PRINT "RECOMMENCEZ ";: GOTO 20
50 PRINT "VOUS AVEZ TAPE 1"
60 END
70 PRINT "VOUS AVEZ TAPE 2"
```

EXPLICATION

Ligne 20 Affectez 1 ou 2 à la variable A.

30 Suivant la valeur de A, vous avez un saut à la ligne 50 ou 70.

40 Si vous avez entré un chiffre qui ne correspond pas à la quantité d'arguments de GOTO, cette ligne est immédiatement exécutée et renvoie à la ligne 20.

50 Si vous avez tapé 1 cette ligne s'exécute.

60 Arrête le programme.

70 Si vous avez tapé 2 cette ligne s'exécute.

OPEN

TRADUCTION

OPEN, ouvre

UTILISATION

OPEN ouvre un fichier.

```
10 REM OPEN
20 OPEN 2,8,1,"EXEMPLE"
30 PRINT#2,"BONJOUR"
40 CLOSE 2
```

EXPLICATION

Ligne 20 Ouvre en écriture, le fichier 2 nommé EXEMPLE, sur le périphérique 8. Le premier argument est le n° de fichier qui sert de référence aux autres commandes. Le deuxième est le n° de périphérique. Le troisième indique l'écriture ou la lecture.

30 Ecrit dans le fichier n°2 la chaîne de caractères entre guillemets. Nous pouvons écrire des contenus de variables entières, numériques ou alphanumériques.

40 Ferme le fichier n°2. Lorsque le travail sur un fichier est terminé ne jamais oublier de le fermer.

NOTA

N° de périphériques	Troisième N°
0 écran	0 lecture
1 cassette	1 écriture
4 imprimante	2 fin de bande
8 disquette	

OR

TRADUCTION

OR, ou.

UTILISATION

OR, cherche si une des deux relations est vraie.

```
 10 REM OR (+)
 20 LET A$="UNE RELATION EST VRAIE"
 30 LET A=5
 40 IF (A)2) OR 9=3 THEN PRINT A$
 50 LET B$="S"
 60 IF "D"(B$ OR ("F"="G") THEN PRINT A$
 70 IF (1=2)+(18)5) THEN PRINT A$
 80 PRINT 1=2 OR 3)=4
 90 PRINT 1=2+3(4

RUN
UNE RELATION EST VRAIE
UNE RELATION EST VRAIE
UNE RELATION EST VRAIE
 0
-1
```

EXPLICATION

Ligne 20 Affecte la chaine de caractères à la variable A$.
30 Affecte la valeur 5 à la variable numérique A.
40 Si A est supérieur à 2, ou 9 égale 3, A$ s'affiche.
50 Affecte la lettre "S", à la variable B$.

60 Si le code de la lettre "D" est inférieur au code de la lettre contenu dans B$, ou si le code de "F" est égal au code de "G", le contenu de A$ s'affiche.
70 Le signe plus remplace OR.
80 Les deux relations sont fausses, donc 0 s'affiche.
90 Une des deux relations est vraie, donc 1 s'affiche.

PARENTHESES "()"

UTILISATION

Les parenthèses délimitent des opérations arithmétiques ou des fonctions.

```
10 REM PARENTHESES ()          RUN
20 PRINT SQR (64)+17           25
30 PRINT SQR (64+17)           9
40 PRINT 4*3+5                 17
50 PRINT 4*(3+5)               32
```

EXPLICATION

Ligne 20 Donne la racine carrée de 64, puis additionne 17.
30 Donne la racine carrée de la somme entre parenthèses. Une opération entre parenthèses est toujours effectuée en premier.
40 Multiplie 4 par 3, puis additionne 5.
50 Additionne 5 à 3, puis multiplie le résultat par 4.

NOTA

Ordre de priorité des opérateurs arithmétiques et logiques.

```
()      priorité 1
**      priorité 2
* et /  priorité 3
+ et -  priorité 4
NOT     priorité 5
AND     priorité 6
OR      priorité 7
```

Lorsque des opérateurs ont la même priorité, les opérations s'effectuent de gauche à droite.

PEEK

TRADUCTION

PEEK, regarder à.

UTILISATION

PEEK donne la valeur décimale rangée à l'adresse précisée.

```
10 REM PEEK                          RUN
20 POKE 32390,40                      40
30 PRINT PEEK (32390)                A
40 POKE 32390,65                      65
50 PRINT CHR$ (PEEK (32390))         A
60 LET A=PEEK (32390)
70 PRINT A
80 PRINT CHR$ (A)
```

EXPLICATION

Ligne 20 Place à l'adresse 32390, la valeur décimale 40.
30 Affiche le contenu de l'adresse 32390.
40 Place à l'adresse 32390, la valeur décimale 65.
50 Affiche le caractère, correspondant au code ASCII placé à l'adresse 32390.
60 La lecture d'une adresse peut être affectée à une variable numérique.
70 Affiche le contenu de la variable numérique A.
80 Affiche le caractère ASCII rangé à l'adresse 32390.

POINT "."

UTILISATION

Le point remplace la virgule dans un nombre décimal.

```
 10 REM POINT
 20 PRINT 12.345*2
 30 PRINT 12,345*2

RUN
24.69
12          690
```

EXPLICATION

Ligne 20 Le nombre 12,345 est multiplié par 2.

30 Affiche le nombre 12, la virgule donne une tabulation de 10 colonnes, puis 345 est multiplié par 2.

POINT-VIRGULE ";"

UTILISATION

Le point-virgule sépare des chaînes ou des variables.

```
10 REM POINT_VIRGULE (;)
20 LET A=45: LET B=6
30 PRINT A;"*";B;"=";A*B
40 LET A$="COMMODORE": LET B=64
50 PRINT A$;B
60 PRINT "BON";
70 PRINT "JOUR"
```

```
RUN
 45 * 6 = 270
COMMODORE 64
BONJOUR
```

EXPLICATION

Ligne 20 Affecte 45 et 6 aux variables numériques A et B.

30 Affiche sur une seule ligne: le contenu de A, la chaîne de caractères, le contenu de B, la chaîne de caractères, puis le résultat du contenu de A multiplié par le contenu de B. Les points-virgules évitent de passer à la ligne suivante (suppression du retour chariot).

40 Affecte la chaîne de caractères à la variable alphanumérique A$ et le nombre à la variable numérique B.

50 Affiche le contenu de A$ et de B.

60 Affiche la chaîne de caractères "BON".

70 Le point-virgule de la ligne précédente permet l'affichage de JOUR à la suite de BON.

POKE

TRADUCTION

POKE, mettre.

UTILISATION

POKE place une valeur décimale à une adresse précisée.

```
10 REM POKE
20 POKE 32390,40
30 PRINT PEEK (32390)
40 LET A=208
50 POKE 32390,A
60 PRINT PEEK (32390)
70 LET B=32400: LET C=65
80 POKE B,C
90 PRINT CHR$ (PEEK (B))
```

```
RUN
 40
 208
A
```

EXPLICATION

Ligne 20 Place à l'adresse 32390, le nombre décimal 40.
30 Affiche le contenu de l'adresse 32390.
40 Affecte 208 à la variable numérique A.
50 Place à l'adresse 32390, le contenu de A.
60 Affiche le contenu de l'adresse 32390.
70 Affecte 32400 et 65 aux variables numériques B et C.
80 Place le contenu de C, à l'adresse correspondant au contenu de B.
90 Affiche la lettre du code ASCII rangé à l'adresse B.

POS

TRADUCTION

Abréviation de POSITION, position.

UTILISATION

Donne la position du curseur

```
 10 REM POS
 20 A$ = "ANTICONSTITUTIONNELLEMENT"
 30 B$ = "TONNE"
 40 PRINT TAB( 10);B$; POS (0)
 50 PRINT A$; POS (4)
 60 PRINT SPC( 10);B$; POS (255)

RUN
          TONNE 15
ANTICONSTITUTIONNELLEMENT 25
          TONNE 15
```

EXPLICATION

Ligne 20 Affecte la chaîne de caractères à la variable A$.
     30 Affecte la chaîne de caractères à la variable B$.
     40 10 espaces plus un mot de 5 lettres donnent la position 15.
     50 Nous avons la longueur du mot.
     60 L'argument de POS est aléatoire mais nécessaire.

POUR CENT "%"

UTILISATION

Une variable entière garde l'entier le plus proche de la décimale qu'on lui affecte.

```
10 REM POURCENTAGE...%...
20 A = 10 / 3
30 A% = A
40 PRINT A%
50 C% = A * 3
60 PRINT C%
70 PRINT A% * 3
80 PRINT INT (A * 3)
```

```
RUN
 3
 10
 9
 9
```

EXPLICATION

Ligne 20 Affecte le résultat de la division à la variable numérique A.

30 Affecte le contenu de la variable numérique A à la variable entière A%.

40 Affiche le contenu de A%, donc 3 car 3,3333....est plus près de 3.

50 Affecte le contenu de la variable numérique A*3 à la variable entière C%.

60 Affiche le contenu de C%, donc 10 car 9,9999....est plus près de 10.

70 Affiche le contenu de A%*3.

80 Contrairement à la ligne 60, INT donne l'entier inférieur.

PRINT

TRADUCTION

PRINT, imprimer.

UTILISATION

PRINT affiche une chaîne ou le contenu d'une variable.

```
10 REM PRINT
20 PRINT "COM";
30 PRINT "MODORE",
40 LET A=64
50 PRINT A
60 PRINT 45*3+(45/3)
70 PRINT TAB(10)"COLONNE 10"
```

```
RUN
COMMODORE  64
 150
          COLONNE 10
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères "COM". Le point-virgule permettra au prochain affichage de se faire sur cette ligne.

30 Affiche "MODORE" à la suite de "COM". La virgule permettra au prochain affichage de se faire sur cette ligne, mais avec une tabulation de 10 colonnes.

40 Affecte 64 à la variable numérique A.

50 Affiche le contenu de A.

60 Affiche le résultat de l'opération (la division entre parenthèses est effectuée en premier).

70 Affiche à partir de la tabulation 10, la chaîne de caractères entre guillemets.

PRINT

TRADUCTION

PRINT#, écrire.

UTILISATION

PRINT# écrit sur un fichier.

```
10 REM PRINT#
20 OPEN 2,8,1,"EXEMPLE"
30 PRINT#2,"BONJOUR"
40 CLOSE 2
```

EXPLICATION

Ligne 20 Ouvre en écriture, le fichier 2 nommé EXEMPLE, sur le périphérique 8. Le premier argument est le n° de fichier qui sert de référence aux autres commandes. Le deuxième est le n° de périphérique. Le troisième indique l'écriture ou la lecture.

30 Ecrit dans le fichier n°2 la chaîne de caractères entre guillemets. Nous pouvons écrire des contenus de variables entières, numériques ou alphanumériques.

40 Ferme le fichier n°2. Lorsque le travail sur un fichier est terminé ne jamais oublier de le fermer.

NOTA

Voir OPEN.

PUISSANCE "↑"

UTILISATION

↑ élève un nombre à une puissance égale à l'argument.

```
10 REM PUISSANCE (↑)          RUN
20 LET A=5                    25
30 PRINT A↑2                  125
40 PRINT 5↑3                  7
50 PRINT SQR (7↑2)            9
60 PRINT 81↑(1/2)             7
70 PRINT 343↑(1/3)
```

EXPLICATION

Ligne 20 Affecte le chiffre 5 à la variable numérique A.
30 Elève le contenu de A à la puissance 2.
40 Elève 5 à la puissance 3.
50 Extrait la racine carrée de 7 élevée à la puissance 2.
60 Un nombre élevé à la puissance inverse donne la racine de ce nombre.
70 Affiche la racine cubique du nombre 343.

NOTA
Voir priorités.

READ

TRADUCTION

READ, lire.

UTILISATION

READ lit le contenu de DATA.

```
10  REM  READ
20  INPUT "TAPEZ UN CHIFFRE ENTRE 1 ET 6 :";B
30  IF B ) 6 OR B ( 1 THEN  PRINT "RECOMMENCEZ": GOTO 20
40  FOR L = 1 TO B
50  READ K
60  DATA  100,200,300,400,500,600
70  NEXT L
80  PRINT K
```

EXPLICATION

Ligne 20 Affectez un nombre à la variable B.

30 Si le nombre n'est pas dans le créneau 1 à 6, retour ligne 20.

40 Affecte 1 à B, à la variable de contrôle L.

50 Lecture des données du tableau K.

60 Affecte les données au tableau K, (10 maximum voir DIM).

70 Retour ligne 50 un nombre de fois équivalent au contenu de B.

80 Affiche la dernière donnée lue.

REM

TRADUCTION

Abréviation de REMARK, remarque.

UTILISATION

REM insère un commentaire dans un programme.

```
 10 REM REM
 20 REM CE PROGRAMME
 30 REM NE FAIT RIEN
 40 PRINT "REM NE FAIT RIEN"

RUN
REM NE FAIT RIEN
```

EXPLICATION

Lignes 10 à 30 Commentaires n'ayant aucune conséquence sur le déroulement du programme.

40 Affiche la chaîne de caractères entre guillemets.

RENAME DOS

TRADUCTION

RENAME, renommer.

UTILISATION

Change le nom d'un fichier.

```
OPEN 15,8,15
PRINT# 15,"RENAME:new=old"
P shift R 15,"R:new=old"
CLOSE 15
```

EXPLICATION

Ligne 1 Ouvre le fichier 15 du périphérique 8, le dernier argument est le numéro de canal.
2 Sur votre disquette, old sera renommé new.
3 Idem ligne 2, mais en abrégé.
4 Ferme le fichier 15.

RESTORE

TRADUCTION

RESTORE, restaurer.

UTILISATION

RESTORE permet de relire des données en partant du début.

```
 10 REM RESTORE
 20 INPUT "TAPEZ UN NOMBRE ENTRE UN ET SEPT :";J
 30 IF J > 7 OR J < 1 THEN  PRINT "MAUVAIS": GOTO  20
 40 FOR I = 1 TO J
 50 READ J$
 60 DATA LUNDI,MARDI,MERCREDI,JEUDI,VENDREDI,SAMEDI,DIMANCHE
 70 NEXT I
 80 PRINT J$
 90 RESTORE
100 GOTO 20
```

EXPLICATION

Ligne 20 Affectez un nombre à la variable J.

30 Si le nombre n'est pas dans le créneau 1 à 7, retour ligne 20.

40 Affecte 1 à J, à la variable de contrôle I.

50 Lecture des données du tableau J$.

60 Affecte les données au tableau J$ (10 maximum voir DIM).

70 Retour ligne 50 un nombre de fois équivalent au contenu de J.

80 Affiche la dernière donnée lue.
90 Permet de reprendre la lecture des données à leur début (essayez le programme en supprimant cette ligne).
100 Le programme recommence (pour arrêter, appuyez sur STOP).

RETURN

TRADUCTION

RETURN, retour.

UTILISATION

RETURN renvoie à la ligne qui suit le dernier GOSUB rencontré.

```
10 REM RETURN                    RUN
20 PRINT "LIGNE 20"              LIGNE 20
30 GOSUB 80                      SOUS PROGRAMME
40 PRINT "LIGNE 40"              LIGNE 40
50 GOSUB 80                      SOUS PROGRAMME
60 PRINT "LIGNE 60"              LIGNE 60
70 END
80 PRINT "SOUS PROGRAMME"
90 RETURN
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères entre guillemets.
30 Envoie au sous-programme débutant à la ligne 80.
40 Affiche la chaîne de caractères entre guillemets.
50 Envoie au sous-programme débutant à la ligne 80.
60 Affiche la chaîne de caractères entre guillemets.
70 Stoppe le programme.
80 Affiche la chaîne de caractères entre guillemets.
90 Renvoie à la ligne qui suit le dernier GOSUB rencontré.

NOTA

RETURN est obligatoire à la fin d'un sous-programme.

RIGHT $

TRADUCTION

RIGHT, droite.

UTILISATION

Extrait un certain nombre de caractères droits d'une chaîne.

```
10 REM RIGHT$
20 X$ = "COMMODORE 64 "
30 PRINT RIGHT$ (X$,(3+4) / 2); LEFT$ (X$,6);"ES"

RUN
64 COMMODES
```

EXPLICATION

Ligne 20 Affecte la chaîne de caractères à la variable X$.

30 Affiche: les 3 derniers caractères de X$ (lorsque l'argument contient des décimales, il n'est retenu que l'entier), les 6 premiers caractères de x$, puis le mot ES.

NOTA

L'argument peut contenir des opérations arithmétiques et des variables numériques.

RND

TRADUCTION

Abréviation de RANDOM, au hasard.

UTILISATION

RND donne un nombre au hasard.

```
10 REM RND
20 PRINT INT ( RND (1) * 10) + 1
30 PRINT RND (1)
40 LET A=INT ( RND (1) * 6)
50 PRINT A+100
```

EXPLICATION

Ligne 20 Le plus grand entier inférieur au nombre aléatoire entre 0 et 1 multiplié par 10 est extrait, puis 1 est additionné à ce résultat. Nous avons donc un nombre au hasard entre 1 et 10.

30 Affiche un nombre aléatoire entre 0 et 1.

40 Affecte à la variable numérique A, le plus grand entier inférieur au nombre aléatoire entre 0 et 1 multiplié par 6.

50 Affiche le résultat de 100 plus contenu de A. Nous avons donc un nombre au hasard entre 100 et 105.

NOTA

RND est une séquence fixe de 65535 nombres mélangés entre 0 et 1. Les nombres semblent aléatoires car la séquence démarre à des endroits différents.

RUN

TRADUCTION

RUN, courir.

UTILISATION

RUN lance l'exécution d'un programme.

```
10 REM RUN
20 PRINT "LIGNE 20"
30 LET A=3
40 LET A$="VARIABLE"
50 PRINT A$;A
60 FOR I=1 TO 1000:NEXT
70 RUN 80
80 PRINT A$
90 PRINT A
```

```
RUN
LIGNE 20
VARIABLE 3

 0
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères entre guillemets.
30 Affecte 3 à la variable numérique A.
40 Affecte la chaîne de caractères entre guillemets à la variable alphanumérique A$.
50 Affiche le contenu de A$ et de A.

60 Boucle d'attente d'une seconde.
70 RUN 80 lance le déroulement du programme à partir de la ligne 80.
80 Affiche le contenu de la variable A$ (vide).
90 Affiche le contenu de A (0).

NOTA
RUN vide les variables alphanumériques et met à 0 les variables numériques.

SAVE

TRADUCTION

SAVE, sauver.

UTILISATION

SAVE enregistre un programme sur cassette.

```
 10 REM SAVE
 20 PRINT "DANS TROIS SECONDES"
 30 PRINT "CE PROGRAMME SE CHARGERA"
 40 PRINT "SUR CASSETTE"
 50 PRINT "SOUS LE NOM DE: EXEMPLE"
 60 FOR I=1 TO 3000:NEXT
 70 PRINT"CLR/HOME"
 80 SAVE "EXEMPLE"
 90 PRINT TAB(9);"C'EST FINI"
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères entre guillemets.
30 Affiche la chaîne de caractères entre guillemets.
40 Affiche la chaîne de caractères entre guillemets.
50 Affiche la chaîne de caractères entre guillemets.
60 Immobilise le programme pendant trois secondes.
70 Efface l'écran.
80 Ce programme se sauvegarde sur cassette, sous le nom de EXEMPLE.
90 Affiche à la colonne 9, la chaîne de caractères entre guillemets.

NOTA

Sur disquettes faire: SAVE"exemple",8.

SAVE DOS

TRADUCTION

SAVE, sauver.

UTILISATION

Sauvegarde un programme.

```
SAVE"name",8
S shift A"name",8
SAVE"0:name",8
```

EXPLICATION

Ligne 1 Le programme name s'enregistre sur le périphérique 8 (la disquette).

2 Idem ligne 1, mais en abrégé.

3 Sauvegarde le nouveau programme name en supprimant l'ancien.

SCRATCH DOS

TRADUCTION

Abréviation de SCRATCH OUT, effacer.

UTILISATION

Supprime un fichier.

```
OPEN 15,8,15
PRINT# 15,"SCRATCH:name"
P shift R 15,"S:name"
CLOSE 15
```

EXPLICATION

Ligne 1 Ouvre le fichier 15 du périphérique 8, le dernier argument est le numéro de canal.
2 Name sera supprimé de votre disquette.
3 Idem ligne 2, mais en abrégé.
4 Ferme le fichier 15.

SGN

TRADUCTION

Abréviation de SIGN, signe.

UTILISATION

SGN donne le signe d'un argument.

```
 10 REM SGN
 20 PRINT SGN (24)*3
 30 PRINT SGN (3-3)
 40 PRINT SGN (34-45)
 50 LET A=2*18
 60 PRINT SGN (A)
 70 LET B=SGN (45-50)
 80 PRINT B

RUN
 3
 0
-1
 1
-1
```

EXPLICATION

Ligne 20 Donne le signe du nombre 24 puis le multiplie par 3, (le signe de 24 est 1, 1*3=3, donc le chiffre 3 s'affiche).

30 Donne le signe du résultat de l'opération entre parenthèses.

40 Donne le signe du résultat de l'opération entre parenthèses.
50 Affecte le résultat de la multiplication à la variable numérique A.
60 Donne le signe du contenu de A.
70 Affecte à la variable numérique B, le signe du résultat de la soustraction entre parenthèses.
80 Affiche le contenu de B.

NOTA
1 représente un chiffre positif.
-1 représente un chiffre négatif.
0 représente un chiffre nul.

SIN

TRADUCTION

Abréviation de SINE, sinus.

UTILISATION

SIN donne le sinus d'un argument exprimé en radians.

```
10 REM SIN
20 PRINT SIN (2)
30 PRINT SIN (2)*4-5
40 LET B=SIN (2*4-5)
50 PRINT B
60 LET A=1
70 PRINT SIN (A)
80 PRINT SIN (A+2)
```

```
RUN
 .909297427
-1.36281029
 .141120008
 .841470985
 .141120008
```

EXPLICATION

Ligne 20 Affiche le sinus du nombre entre parenthèses.

30 Donne le sinus du premier nombre, multiplie le résultat par 4, puis soustrait 5.

40 Le sinus de l'argument entre parenthèses est affecté à la variable B.

50 Affiche le contenu de la variable B.

60 Le chiffre 1 est affecté à la variable A.

70 Affiche le sinus du contenu de la variable A.

80 L'argument entre parenthèses peut contenir des variables, des nombres et des opérations arithmétiques.

SOUSTRACTION "-"

UTILISATION

Le signe (-) soustrait l'élément de droite à celui de gauche.

```
10 REM SOUSTRACTION (-)          RUN
20 LET A=19-3                    16
30 PRINT A                       44
40 LET B=60                      4
50 PRINT B-A                     18
60 LET C=A*4-B
70 PRINT C
80 PRINT 21-3
```

EXPLICATION

Ligne 20 Affecte le résultat de l'opération, 19 moins 3, à la variable numérique A.

30 Affiche le contenu de la variable A.

40 Affecte le nombre 60, à la variable numérique B.

50 Affiche le résultat de: contenu de B moins contenu de A.

60 Affecte à la variable numérique C, le résultat de l'opération: contenu de A multiplié par 4 moins contenu de B.

70 Affiche le contenu de C.

80 Affiche le résultat de l'opération 21 moins 3.

NOTA

Voir PARENTHESES (priorités).

SPC

TRADUCTION

Abréviation de SPACE, espace.

UTILISATION

Insère des espaces

```
10 REM SPC
20 PRINT SPC(2*5)"UN"
30 PRINT SPC(9)"UN"; SPC(9)"DEUX"
40 PRINT TAB(10)"TAB"
50 PRINT SPC(10)"SPC"

RUN
          UN
         UN         DEUX
          TAB
          SPC
```

EXPLICATION

Ligne 20 L'argument peut contenir une opération arithmétique. Le résultat est 10, nous avons donc 10 espaces entre le bord gauche et l'affichage.

30 Nous avons 9 espaces entre le bord gauche et UN, puis 9 autres entre UN et DEUX.

40 Nous avons 10 espaces entre le bord gauche et TAB.

50 Nous avons 10 espaces entre le bord gauche et SPC.

SQR

TRADUCTION

Abréviation de SQUARE, carré.

UTILISATION

SQR extrait la racine carrée.

```
 10 REM SQR                      RUN
 20 PRINT SQR (9)                 3
 30 LET A=1089                    33
 40 PRINT SQR (A)                 8
 50 PRINT SQR ((2+6)*8)           27
 60 PRINT SQR (9)*9              -156
 70 PRINT SQR (A)-189             30
 80 PRINT SQR (A-189)             7
 90 LET B=SQR (49)
100 PRINT B
```

EXPLICATION

Ligne 20 Affiche la racine carrée de 9.
30 Affecte 1089 à la variable numérique A.
40 Affiche la racine carrée du contenu de A.
50 Affiche la racine carrée du résultat de l'opération entre parenthèses.
60 Donne la racine carrée de 9, puis multiplie le résultat par 9.
70 Donne la racine carrée de A, puis soustrait 189.
80 Affiche la racine carrée du contenu de A moins 189.
90 Affecte la racine carrée de 49 à la variable numérique B.
100 Affiche le contenu de la variable numérique B.

STEP

TRADUCTION

STEP, pas.

UTILISATION

STEP donne le pas d'incrémentation dans une boucle FOR...TO...NEXT.

```
 10 REM STEP
 20 FOR A = 1 TO 10 STEP (4 - 2)
 30 PRINT A
 40 NEXT A
 50 INPUT "TAPEZ 0 OU 3.5 :";X
 60 FOR B = 1 TO 10 STEP X
 70 PRINT B
 80 NEXT B
 90 INPUT "TAPEZ -2 OU -1 :";Z
100 FOR C = 10 TO 0 STEP Z
110 PRINT C
120 NEXT C
```

EXPLICATION

Ligne 20 STEP incrémente la boucle de 2 (l'argument peut être une opération).

30 A chaque passage, la nouvelle valeur de A s'affiche.

40 Retour ligne 30 jusqu'à ce que la variable de contrôle A soit à sa valeur maximum.

Lignes 50 à 80 L'argument de STEP peut être une variable. 0 bloque l'incrémentation. Une incrémentation peut contenir des décimales.

90 à 120 Un argument négatif permet une décrémentation.

STOP

TRADUCTION

STOP, arrêt.

UTILISATION

STOP provoque une interruption dans le déroulement d'un programme.

```
10 REM STOP
20 PRINT "APPUYEZ SUR STOP"
30 FOR I=1 TO 500: NEXT
40 GOTO 20
```

EXPLICATION

Ligne 20 Affiche la chaîne de caractères.
30 Boucle d'attente d'une demi-seconde.
40 Retour ligne 20.

NOTA

La boucle 20 à 40 s'exécutera toutes les demi-secondes. Pour l'arrêter vous devrez appuyer sur la touche STOP.

STR$

TRADUCTION

Abréviation de STRING, chaîne.

UTILISATION

STR$ convertit une valeur numérique en chaîne alphanumérique.

```
10 REM STR$
20 LET A=123
30 PRINT STR$(A)
40 LET A$=STR$(A)
50 PRINT LEN(A$)
60 LET B$="LE NOMBRE EST"
70 PRINT B$+A$
```

```
RUN
 123
 3
LE NOMBRE EST 123
```

EXPLICATION

Ligne 20 Affecte le nombre à la variable numérique A.

30 Avant d'être affiché, le contenu de la variable numérique A est converti en variable alphanumérique.

40 Affecte la variable numérique A à la variable alphanumérique A$.

50 Donne la longueur de la chaîne contenue dans la variable alphanumérique A$.

60 Affecte la chaîne de caractères entre guillemets à la variable alphanumérique B$.

70 Nous avons une concaténation des chaînes, A$ et B$.

SUPERIEUR ">"

UTILISATION

Permet de comparer deux valeurs numériques ou deux chaînes de caractères.

```
 10 REM SUPERIEUR
 20 LET A=6
 30 IF A)5 THEN PRINT "6 ) 5"
 40 LET A$="C"
 50 IF "D")A$ THEN PRINT "D APRES C"

RUN
6 ) 5
D APRES C
```

EXPLICATION

Ligne 20 Affecte le chiffre 6 à la variable numérique A.

30 Si le contenu de la variable numérique A est supérieur à 5, la chaîne de caractères entre guillemets s'affiche.

40 Affecte la lettre "C" à la variable alphanumérique A$.

50 Si le code de la lettre "D", est supérieur au code de la lettre contenue dans la variable A$, la chaîne de caractères s'affiche.

SUP. OU EGAL ">="

UTILISATION

Permet de comparer deux valeurs numériques ou deux chaînes de caractères.

```
10 REM SUPERIEUR OU EGAL
20 LET A=4
30 IF A)=3 THEN PRINT "4 ) 3"
40 IF A)=4 THEN PRINT "A EGALE 4"
50 LET A$="G"
60 IF "J")=A$ THEN PRINT "J APRES G"
70 IF "G")=A$ THEN PRINT "A$ EGALE G"
```

```
RUN
4 ) 3
A EGALE 4
J APRES G
A$ EGALE G
```

EXPLICATION

Ligne 20 Affecte le chiffre 4 à la variable numérique A.

30 Si le contenu de la variable numérique A est supérieur ou égal à 3, la chaîne de caractères entre guillemets s'affiche.

40 Si le contenu de la variable numérique A est supérieur ou égal à 4, la chaîne de caractères entre guillemets s'affiche.

50 Affecte la lettre "G" à la variable alphanumérique A$.

60 Si le code de la lettre "J", est supérieur ou égal au code de la lettre contenue dans la variable A$, la chaîne de caractères s'affiche.

70 Si le code de la lettre "G", est supérieur ou égal au code de la lettre contenue dans la variable A$, la chaîne de caractères s'affiche.

SYS

TRADUCTION

Abréviation de SYSTEM, système.

UTILISATION

Lance l'exécution d'un programme en langage machine.

EXPLICATION

SYS suivi d'un nombre décimal ou d'une variable numérique dans le créneau 0 à 65535, lance à cette adresse l'exécution d'un sous-programme en langage machine.
EX: SYS 64738 vide complètement la mémoire, contrairement à NEW qui n'efface que le programme Basic (ici nous appelons un sous-programme fait par le constructeur).
Avec la commande SYS, vos sous-programmes en langage machine pourront être appelés à partir du Basic.

TAB

TRADUCTION

Abréviation de TABULATOR, tabulateur.

UTILISATION

TAB définit l'endroit où commencera l'affichage.

```
10 REM TAB                          RUN
20 PRINT TAB(40);"LIGNE 20"
30 PRINT TAB(25);"PAGE";TAB(30);1  LIGNE 20
40 PRINT TAB(6*2);"MILIEU"                                  PAGE 1
50 LET A=12                                      MILIEU
60 PRINT TAB(A);"CENTRE"                         CENTRE
70 PRINT ,"COLONNE 11"                           COLONNE 11
```

EXPLICATION

Ligne 20 La tabulation 40 fait commencer l'affichage au début de la ligne suivante.

30 Affiche "PAGE" à la tabulation 25 et "1" à la tabulation 30.

40 L'argument de TAB peut être une opération arithmétique.

50 Affecte 12 à la variable numérique A.

60 L'argument de TAB peut être une variable numérique.

70 La virgule effectue une tabulation à la onzième colonne.

TAN

TRADUCTION

Abréviation de TANGENT, tangente.

UTILISATION

TAN calcule la tangente d'un argument exprimé en radians.

```
10 REM TAN                              RUN
20 PRINT TAN (1.1071487)                 1.99999991
30 LET B=TAN (1.1071487+0.1418971)       3.00000028
40 PRINT B                               .999999992
50 LET A=0.78539816                      3.00000028
60 PRINT TAN (A)                         3
70 PRINT TAN (A+0.46364764)
80 PRINT TAN (ATN(3))
```

EXPLICATION

Ligne 20 Affiche la tangente de l'argument 1.1071487.

30 La tangente du résultat de l'addition entre parenthèses est affectée à la variable B.

40 Affiche le contenu de la variable B.

50 Le nombre est affecté à la variable A.

60 Affiche la tangente du contenu de la variable A.

70 L'argument entre parenthèses peut contenir: des variables, des nombres et des opérations arithmétiques.

80 La tangente et l'arc tangente sont des fonctions mutuellement inverses.

THEN

TRADUCTION

THEN, alors.

UTILISATION

Lorsqu'une condition est satisfaite, l'instruction qui suit THEN s'effectue.

```
10 REM THEN                                RUN
20 FOR A=1 TO 10                           A=1
30 IF A=2 THEN LET B=A+100                 (Attente 2 secondes)
40 IF A=3 THEN FOR I=1 TO 2000:NEXT        B=102
50 IF A=1 THEN PRINT "A=1"
60 NEXT A
70 PRINT "B=";B
```

EXPLICATION

Ligne 20 Affecte 1 à 10 à la variable de contrôle A.

30 Lorsque A égale 2, THEN permet d'affecter le contenu de A+100 à la variable B.

40 Lorsque A égale 3, THEN permet une pause de 2 secondes.

50 Lorsque A égale 1, THEN permet l'affichage de la chaîne de caractères entre guillemets.

60 Retour ligne 30 jusqu'à ce que la variable de contrôle A soit à sa valeur maximum.

70 Affiche la chaîne de caractères entre guillemets et le contenu de B.

TO

TRADUCTION

TO, à.

UTILISATION

TO donne la valeur maximum de la variable de contrôle.

```
 10 REM TO                          RUN
 20 FOR A=1 TO 3                     1
 30 PRINT A                          2
 40 NEXT A                           3
 50 FOR B=1 TO 4*5-22 STEP -1        1
 60 PRINT B                          0
 70 NEXT B                          -1
 80 LET D=4                         -2
 90 FOR C=3 TO D                     3
100 PRINT C                          4
110 NEXT C
```

EXPLICATION

Ligne 20 TO donne la valeur maximum de la variable de contrôle A.

30 Affiche à chaque passage la nouvelle valeur de A.

40 Renvoie ligne 30 jusqu'à ce que le contenu de la variable A soit égal au nombre qui suit TO.

50 L'argument qui suit TO peut être une opération arithmétique.

60 Affiche à chaque passage la nouvelle valeur de B.

70 Renvoie ligne 60 jusqu'à ce que le contenu de la variable B soit égal au résultat de l'opération qui suit TO.
80 Affecte 4 à la variable numérique D.
90 L'argument qui suit TO peut être une variable numérique.
100 Affiche à chaque passage la nouvelle valeur de C.
110 Renvoie ligne 100 jusqu'à ce que le contenu de la variable C soit égal au contenu de la variable qui suit TO.

USR

TRADUCTION

Abréviation de USER, utilisateur.

UTILISATION

USR envoie un argument dans un programme en langage machine.

EXPLICATION

USR(x) L'argument x est envoyé dans un sous-programme en langage machine. La première adresse du sous-programme doit être mise dans les cases mémoire 1 ou 2.

VAL

TRADUCTION

Abréviation de VALUE, valeur.

UTILISATION

VAL convertit une chaîne alphanumérique en valeur numérique.

```
10 REM VAL
20 LET A$="10 CHIENS"
30 LET A=VAL (LEFT$(A$,2))
40 PRINT A*2;RIGHT$(A$,6)
50 PRINT VAL ("13")*2
60 LET B$="SQR 81"
70 PRINT B$;"= ";
80 PRINT SQR (VAL (RIGHT$(B$,2)))
```

```
RUN
 20 CHIENS
 26
SQR 81 = 9
```

EXPLICATION

Ligne 20 Affecte la chaîne de caractères entre guillemets à la variable alphanumérique A$.

30 Les 2 premiers caractères de la variable alphanumérique A$ sont transformés en variable numérique A.

40 Affiche sur une ligne: le contenu de A multiplié par 2, puis les 6 derniers caractères de A$.

50 Pour être multipliée par 2, la chaîne de caractères "13" est transformée en valeur numérique.

60 Affecte la chaîne de caractères entre guillemets à la variable alphanumérique B$.

70 Affiche le contenu de B$ et la chaine entre guillemets. Le point-virgule final permettra au prochain affichage de se faire sur cette ligne.
80 Les 2 derniers caractères de B$ sont transformés en valeur numérique, 9 s'affiche donc (racine carrée de 81 égale 9).

VALIDATE DOS

TRADUCTION

VALIDATE, valider.

UTILISATION

Réorganise la disquette.

```
OPEN 15,8,15
PRINT# 15,"VALIDATE"
P shift R 15,"V"
CLOSE 15
```

EXPLICATION

Ligne 1 Ouvre le fichier 15 du périphérique 8, le dernier argument est le numéro de canal.

2 Après plusieurs effacements et enregistrements, les fichiers occupent plus de secteurs qu'ils ne devraient. Par exemple, lorsqu'un fichier de 5 blocs se substitue à un autre de 10 blocs, le nouveau fichier occupe autant de place que l'ancien. VALIDATE réorganise les fichiers afin qu'ils emplissent l'endroit qui leur est alloué, ce qui a pour effet de libérer une partie de la disquette.

3 Idem ligne 2, mais en abrégé.

4 Ferme le fichier 15.

NOTA

Le temps de validation est assez long, environ une minute et demie.

VERIFY

TRADUCTION

VERIFY, verifie.

UTILISATION

Compare le programme en mémoire à son enregistrement.

```
10  REM VERIFY
20  REM ENREGISTREZ AUPARAVANT CE PROGRAMME
30  INPUT "VOULEZ VOUS VERIFIER  LE PROGRAMME VAL ";A$
40  IF A$ = "OUI" THEN  GOTO 60
50  STOP
60  VERIFY "VAL"
```

EXPLICATION

Ligne 20 Enregistrez ce programme sous le nom de VAL.
30 Tapez OUI ou NON.
40 Si vous avez tapé OUI saut à la ligne 60.
50 Si cette ligne est exécutée le programme s'arrête.
60 Cherche sur cassette si une erreur existe entre le programme résidant et le programme enregistré.

NOTA

Avec unité de disquettes faire: VERIFY "VAL",8.

VIRGULE ","

UTILISATION

La virgule sépare divers éléments.

```
10 REM VIRGULE
20 LET A=3
30 PRINT "TROIS",A
40 PRINT TAB(11);A
50 PRINT TAB(11),A
60 POKE 32390,21
70 PRINT PEEK (32390)

RUN
TROIS       3
            3
                        3
21
```

EXPLICATION

Ligne 20 Affecte le chiffre 3 à la variable numérique A.

30 Affiche la chaîne de caractères "TROIS", puis à la onzième colonne le contenu de A, (la virgule correspond à un TAB 11).

40 Le contenu de A s'affichera à la onzième colonne.

50 Le contenu de A s'affichera à la colonne 22 car TAB 11 et la virgule agissent comme un TAB 22.

60 La virgule sépare les deux arguments de POKE.

70 Affiche le contenu de l'adresse 32390.

WAIT

TRADUCTION

WAIT, attendre.

UTILISATION

WAIT suspend le déroulement d'un programme.

EXPLICATION

WAIT x,y,z

x Adresse mémoire où est arrivé le programme basic. Un caractère devra être placé à cette adresse.

z L'ordinateur fait le OU exclusif entre le contenu de x et z.

y L'ordinateur fait le ET entre y et le résultat précédent. Si le résultat est 0, on recommence l'opération sinon on passe à la ligne suivante.

Achevé d'imprimer
sur les presses
de l'Imprimerie Marcel Bon
70001 Vesoul
Dépôt légal juin 1984
N° d'éditeur : 425
N° d'imprimeur : 2786

MICRO SYSTEMES

ETSF

PASSEPORT POUR COMMODORE 64

Très pratique, cet ouvrage vous présente tous les mots clés du Basic du Commodore 64 dans l'ordre alphabétique. Chaque fonction, instruction ou commande est accompagnée **d'un programme et d'explications détaillées.** Excellent complément du manuel pour les débutants, il est aussi très utile au programmeur pour retrouver rapidement l'emploi d'une instruction.

Méthode de présentation :

Classement alphabétique des commandes, fonctions et instructions, avec, pour chacune d'elles :

- la traduction anglais/français
- son utilisation
- un exemple de programme
- une explication détaillée

ISBN 2-85535-081-6

TARIF CODE B